United States Military Academy
West Point, New York 10996

AN ENLISTED PERSONNEL PERMANENT CHANGE OF STATION (PCS) POLICY ANALYSIS MODEL

OPERATIONS RESEARCH CENTER
TECHNICAL REPORT 96-10-1

Major Michael J. Kwinn, Jr
Analyst, Operations Research Center

Lieutenant Colonel David A. Thomas
Academy Professor, Department of Systems Engineering

Colonel James L. Kays, Ph.D.
Professor and Head, Department of Systems Engineering

15 August 1996

The Operations Research Center is supported by the Assistant Secretary of the Army (Financial Management)

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 0704-0188*

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE 15 August 1996	3. REPORT TYPE AND DATES COVERED Technical Report

4. TITLE AND SUBTITLE An Enlisted Personnel Permanent Change of Station (PCS) policy analysis model	5. FUNDING NUMBERS
6. AUTHOR(S) MAJ Michael Kwinn, LTC David Thomas, COL James Kays	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) USMA Operations Research Center, West Point, New York 10996	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSORING / MONITORING AGENCY REPORT NUMBER 96-10-1
11. SUPPLEMENTARY NOTES	
12a. DISTRIBUTION / AVAILABILITY STATEMENT Distribution Statement A. Approved for public Release; distribution unlimited.	12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

The purpose of this research is to develop an analytical capability that allows personnel managers and analysts at the US Army Personnel Command (PERSCOM) to rapidly assess impact of policy change. This first effort provides a "proof of principle" discrete event simulation of a crucial aspect of personnel policy and management: the permanent change of station or PCS move. PCS moves cost the Department of Defense over $1 billion annually; these moves dramatically affect overall readiness of our forces. Current models are static and based on steady-state assumptions that are no longer valid in a Army that continues to downsize. Current models require weeks of lead time to effect policy analysis. Analysts at PERSCOM must often rely on heuristic and ad hoc approaches to address proposed policy change.

In this "proof of principle", we focus on one military occupational skill (MOS), the Air Defense Patriot Crewmember, and model PCS moves from continental and out-of-continent moves, for both training (TDA) units and operational (TOE) units. We further stratify the analysis by considering the following events: promotion, demotions, attrition, orders processing, and actual movement. ProModel software provides the platform for model development and implementation.

This model will allow personnel managers to quickly modify Time-on-Station requirements or fill priorities. The managers can then determine the effects of such changes by comparing the results of the simulation runs with those of the base case runs. Preliminary results indicate that the simulation model adequately represents both promotions and PCS moves at five installations and for the total TDA units.

14. SUBJECT TERMS Permanent Change of Station (PCS)			15. NUMBER OF PAGES 150
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED	18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED	19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED	20. LIMITATION OF ABSTRACT

AN ENLISTED PERSONNEL PERMANENT CHANGE OF STATION (PCS) POLICY ANALYSIS MODEL

Lead Analyst
Major Michael J. Kwinn, Jr
Analyst, Operations Research Center

A TECHNICAL REPORT
OF THE
OPERATIONS RESEARCH CENTER
UNITED STATES MILITARY ACADEMY

Directed by
Lieutenant Colonel David A. Thomas
Academy Professor, Department of Systems Engineering

Approved by
Colonel James L. Kays, Ph.D.
Professor and Head, Department of Systems Engineering

15 August 1996

The Operations Research Center is supported by the Assistant Secretary of the Army (Financial Management)

The sponsor for this project is Plans and Analysis Division, EPMD, PERSCOM

About the Author

MAJOR MICHAEL J. KWINN, JR. is an Analyst in the Operations Research Center at the United States Military Academy (USMA), West Point, New York. He received a BS from USMA in 1984, and an MS in Systems Engineering from the University of Arizona. The University of Arizona has accepted him into their Ph.D. program. His research interests are personnel management systems and force structure analysis through the use of optimization and simulation models.

Acknowledgments

I would like to acknowledge the Office of the Assistant Secretary of the Army for Financial Management (ASA-FM) for funding this project. I would also like to acknowledge the many people who helped in the development of this model:

Lieutenant Colonel David A. Thomas, my senior investigator, for introducing this research to me and allowing me free reign to do this work.

Lieutenant Colonel James Thomas, my client, for quickly providing me with everything I needed to complete this project.

Major Robert Phelan, another Analyst in the ORCEN, for providing me with invaluable technical advice on ProModel.

Captain Doug Hersh, ODCSPER, for providing me with data at a moments notice and technical expertise on the models they run in his office.

Second Lieutenants Edward Stover, Brian Tucker, and James Zopelis, all recent graduates of USMA, who worked with me on this project in their Advanced Individual Study course. Through our discussions, we worked through many problems in the model development. Their programming ability was invaluable.

Table of Contents

Executive Summary

The purpose of this research is to develop an analytical capability that allows personnel managers and analysts at the US Army Personnel Command (PERSCOM) to rapidly assess impact of policy change. This first effort provides a "proof of principle" discrete event simulation of a crucial aspect of personnel policy and management: the permanent change of station or PCS move. PCS moves cost the Department of Defense over $1 billion annually; these moves dramatically affect overall readiness of our forces. Current models are static and based on steady-state assumptions that are no longer valid in a Army that continues to downsize. Current models often require weeks of lead time to effect policy analysis. Analysts at PERSCOM must often rely on heuristic and ad hoc approaches to address proposed policy change.

In this "proof of principle," we focus on one military occupational skill (MOS), the Air Defense Patriot Crewmember, and model PCS moves from continental and out-of-continent moves, for both training (TDA) units and operational (TOE) units. We further stratify the analysis by considering the following events: promotions, demotions, attrition, orders processing, and actual movement. ProModel© software provides the platform for model development and implementation.

This model will allow personnel managers to quickly modify Time-on-Station requirements or fill priorities. The managers can then determine the effects of such changes by comparing the results of the simulation runs with those of the base case runs. Preliminary results indicate that the simulation model adequately represents both promotions and PCS moves at five installations and for the total TDA units.

Additionally, we have achieved useful information within 10-15 minutes which otherwise would have required extensive man-hours of effort.

Implementation to analyze the "test" MOS requires development of user interfaces to tie the simulation model to the Enlisted Master File. This addition will allow the managers to verify the base case scenario. After this development, further validation of the model may require the client to update required rates and parameters to reflect actual personnel system values.

This model should be expanded to include all MOSs in the Army. This would give personnel managers at the highest level the ability to conduct policy analysis using their desktop PCs.

1.0 Introduction

In the early stages of Operation Desert Shield military planners determined one of the major threats to US and Allied forces in Saudi Arabia prior to any offensive action was Iraqi Scud missile attacks. The solution was to send Patriot battalions as soon as possible to attack and destroy the in-bound Scuds. This anti-ballistic missile defense capability had never before been proven in combat conditions. When the first Patriot system engaged, fired and destroyed the first Scud missile over Riyadh, the effectiveness of the system became legendary - and its crew members have been on the move since. Suddenly every country allied with the United States wanted Patriot coverage to protect against TBM threats.

The Patriot weapon system and the soldiers that man it were moving all over the globe. Deployments became the norm instead of the exception. Soldiers who were stationed at Fort Bliss deployed to South Korea with their Patriot Battalion. They would return from that deployment only to be sent to Germany. Shortly after arrival to Germany, they would deploy for six months to Southwest Asia (SWA) with a different battalion. It was not unheard of for solders to be away from their families for 24-30 months out of the last 36. Retention rates were dropping dramatically and morale was at an all time low. The large number of deployments did not decrease with time. When the Army decided to make the battalion in Korea a permanent stationing, it did reduce the deployment time and the uncertainty of which battalion would deploy next (Costello, 1996). It also increased the deployment time for the soldiers from 6 to 12 months.

With the end of the Cold War, there was an almost universal cry for a "peace dividend". In the early 1990's, this "peace dividend" took the form of a massive reduction in the Army's personnel end-strength and a realignment of installations and Patriot Crewmembers were not immune from these changes. At the end of fiscal year 1989, the Army's enlisted end-strength was 658,000 soldiers stationed throughout the world. By the end of fiscal year 1995, the enlisted end-strength was only 422,000 soldiers (Hersh, 1995). According to the Army Chief of Staff, General Dennis Reimer, the Army may have to reduce even further to pay for modernization programs (Reimer Interview, 1995). Army planners are currently analyzing options on how to reduce the Army end-strength by an additional 20,000-40,000 soldiers.

Senior leaders in the US Army Personnel Command (PERSCOM) demand from their branch managers immediate analysis of the almost daily changes to the personnel system. General Reimer, recently stated, "The models that we've had that have worked well during the Cold War don't adjust well to a changing situation...."(Adelsberger, 1995). Currently, personnel managers base many of their decisions on the *output* of the current models. Personnel managers use most models as decision *support* tools as opposed to decision *analysis* tools.

In this paper, we present a simulation model developed for the Plans and Analysis Branch, Enlisted Personnel Management Division, US Army Personnel Services Command (EPMD, PERSCOM). We designed the model to assist personnel managers in analyzing the impact changes to the personnel system. This model allows managers and analysts to modify policies and determine the subsequent impact on personnel stability and unit readiness.

2.0 Background

When a soldier moves from one location to another, it is a Permanent Change of Station, (PCS) move which generally falls into one of five categories: Unit, Operational, Training, Rotational, and Accessions and Separations. The movement of a soldier as a result of a unit's activation or deactivation is considered a "Unit Move". When a soldier moves between one unit between units both within the Continental United States (CONUS), this is an "Operational Move". An exception to this is when the move is to or from a professional development school. This is considered a "Training Move". Whenever a soldier is moved "over water" (ie. From Germany to the United States), it is counted as a "Rotational Move". Moves which result from a soldier either entering or exiting the Army are considered "Accession and Separation Moves".

The costs associated with all of these moves total over $1 Billion annually. Rotational moves and Accession and Separation moves account for over 90% of the PCS costs (Hix, 1995). The number of moves, and therefore the cost, increased dramatically in recent years due to the draw down. The Army closely managed the voluntary separations by soldiers by MOS and grade but not by unit or geographical region. The result was a large increase in operational moves.

Another significant impact on personnel policy was the changing force structure; especially the reduction in the number of soldiers required in Europe. For years, soldiers knew that roughly every other assignment would be overseas, usually to Europe. This was due to the fact that in the 1980's, approximately 30% of the enlisted soldier authorizations in the Army were in Europe. By the end of FY95, the ratio was down to 13%!

This realignment of forces changed the Army's "comfortable" rotational scheme for soldiers moving from overseas to stateside assignments. As the Army deactivated units in Europe, manning requirements there dropped off dramatically. Personnel began to return to CONUS assignments and there was no longer a need for soldiers to rotate to Europe as often. Base closures and threats of even further force reductions created uncertainty for the Army's leaders, too. Each base-closure recommendation from the Base Realignment Commission (BRAC) further disrupted the personnel system.

2.1 Needs Analysis

We met with LTC James Thomas, Chief, Plans and Analysis Branch, Enlisted Distribution Division, Enlisted Personnel Management Directorate, PERSCOM in September of 1995. He explained many of the situations which arise in his office in dealing with enlisted personnel management. One of the many needs that he explained to us was that his superiors often asked his staff to conduct analysis on the affects of changing personnel policies. For example, they would ask about the effects on unit readiness if the Army extended the tour lengths in Germany. They were also very interested in the effects of reducing the number of Operational moves allowed in a given fiscal year.

LTC Thomas explained that he did not have an analytical tool to directly assist his staff in analyzing these types of problems. Personnel analysts relied on long-handed work with a calculator and a pencil. As the Army continues to operate under increasing tight budgets and to realign units, he anticipated that questions such as these would be asked more frequently.

We determined that he needed an analytical tool with which his staff could quickly and easily modify PCS policies and could determine the effects of these policies on the number of moves and the readiness of the units. The model would have to provide accurate results in a short amount of time.

2.2 Related Work

.There have been number of models developed or studies conducted in recent years which are related to this problem. Each of these models or studies address a slightly different perspective on this problem.

The Office of Economic and Manpower Analysis (OEMA) conducted a study recently in which they recommended policy changes designed to reduce the overall cost of PCS moves annually. OEMA is a research cell associated with the Department of Social Sciences at the United States Military Academy at West Point. This study, entitled "PCS Study Policy Execution" dated November 1995, recommended (among other things) that the Army enforce its policy of keeping soldiers in Germany for 36 months. (Some soldiers PCS prior to 36 months in some instances.) This would save a great deal of money in the short run. This study does not address the impact of the policy on unit readiness or on the budget in future years.

Mr. Mike Hix, of the RAND Corporation, recently briefed a study he and his colleagues at RAND they conducted to determine a means to increase assignment stability. Their initial recommendations were to remove the soldiers from OCONUS assignments. This would greatly increase the stability of soldiers in CONUS assignments in light of the fact they would never have to PCS overseas. This is hardly a realistic recommendation for a personnel manager in that the stationing of personnel overseas is a strategic and political decision, not a personnel management decision.

Another work in-progress at the RAND Corporation is begin done by Mr. Herb Shukiar. He has developed a "Steady-State PCS Model". This is a spreadsheet model designed to determine the CONUS tour length in a steady-state system. This is an iterative model which shows some promise. It allows a user to modify tour lengths in OCONUS and other assignments and then determine the impact this decision has on CONUS tour lengths, or stability. This model does not determine the number of moves which result from such policy decision nor identify the impact on individual unit readiness.

There is a simulation model under development at the US Army Artificial Intelligence Center in Washington, DC (named "Blacksmith"). This is a **very** large simulation model designed to "model the behavior of the Army". It models the *entire* Army. The model is in development but due its scope, it is still a long way from complete development. If it ever can be fully developed, this would indeed be a useful tool.

The modeling effort which shows a great deal of promise is one which employs a Systems Dynamics approach to the personnel system. The Department of Systems Engineering at the United States Military Academy, West Point utilizes the Army Personnel System as the platform to teach this new methodology. Several promising results are forthcoming. In this model, faculty and cadets use continuous-time simulation to replicate the entire personnel system. This model would allow a user to determine how a policy change in the area of recruiting would effect retention, for example. This model is currently in the developmental stages.

3.0 Model Development

3.1 Use of Simulation

The personnel system is very stochastic in nature. The number of soldiers promoted in a given month varies greatly. The number of soldiers who leave the service in a given month also varies greatly. For this reason, we wanted any analytical tool which we developed to account for this variability. Since our model would predict an outcome based on a given set of parameters (the current personnel picture) and rates which varied, we could not just provide our decision maker with one simple answer and not address that variability. There is currently no closed form solution techniques available for this current problem. This was one of the major reasons we chose to develop a simulation of the personnel system. Through the use of simulation, we could replicate the stochastic nature of the personnel system while begin able to analyze each policy by running the simulation a number of times. The results of the simulation could easily be combined and analyzed.

The ProModel® software package provides a robust framework for this simulation. This is a discrete-time simulation package where each minute of simulation time represents one month of real time. ProModel® is an object-oriented simulation package which makes programming the simulation quite easily. Additionally, it provides an outstanding use of animation and a user can easily transform the statistical output into graphical format. Displayed in the figure below is the graphics of the simulation. The numbers within the boxes near the locations are the number of soldiers at that location.

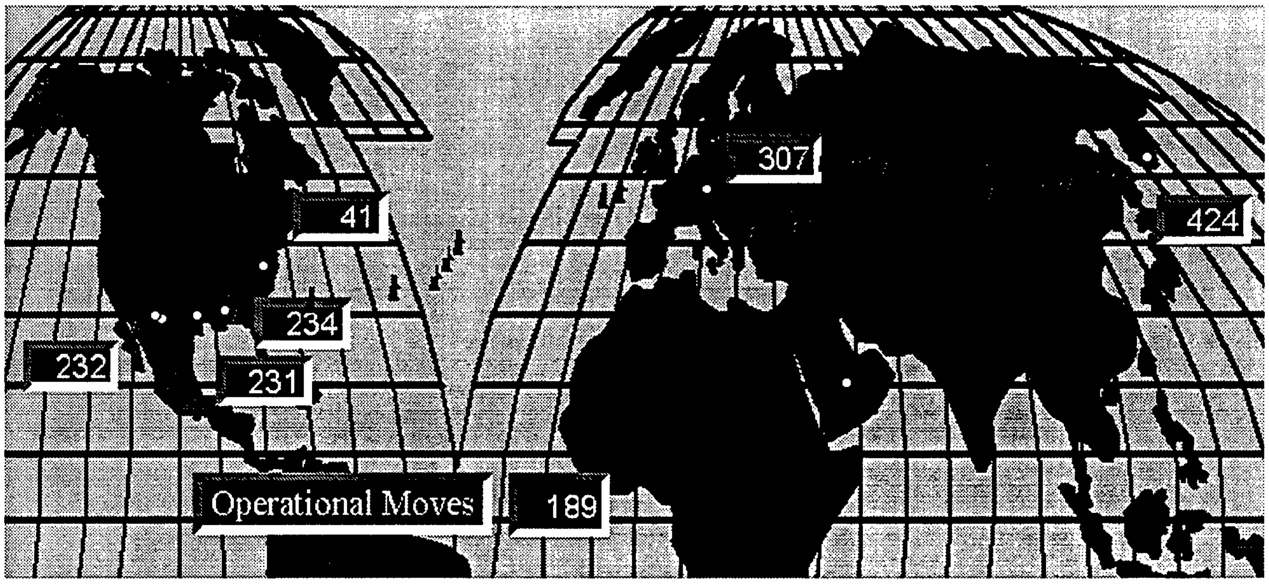

Figure 1: ProModel display for PCS Simulation Model

The animation and the statistical analysis features of the software are extremely valuable when attempting to convince a decision maker on a proposal. However such features can potentially limit the overall size of the model. We will address this later.

3.2 Proof of Principle

Simulating every MOS in the Army is a very large undertaking. The overall modeling effort would take a great deal of work. It would also require a large amount of memory unless we trade-off resolution for aggregation. Prior to launching a large modeling effort, we decided to ensure our efforts would meet the needs of our client. For this reason, the simulation model we develop in this report is a “proof of principle” using only one MOS.

We chose to model 16T, Patriot Crewmember. We decided on this MOS for a number of reasons. First, it was a representative MOS in that a 16T could be stationed in a number of locations both stateside and overseas and could also be assigned to duties away from a Patriot unit, like Drill Sergeant. Second, all of the analysts were Air Defense officers so there was a level of familiarity with the 16T career patterns. Finally, we conducted an interim brief on the project to MG Costello, Chief, Air Defense Branch. He asked that we focus on this MOS due to the lack of assignment stability for these soldiers in light of the requirement for a rotating battalion to deploy to Southwest Asia (SWA).

3.3 Concept for Modularity

Since we were only working on one MOS, we wanted to make the model as modular as possible to allow for easy expansion to encompass all MOSs. This would also allow for adaptability to different policies or realignment of forces. This modular concept also made the initial model development a great deal easier and quicker.

3.3.1 Types of Locations

We categorized each location, or unit assignment, a 16T could be assigned to, into one of three types of locations: CONUS TOE, CONUS TDA and OCONUS. A CONUS TOE unit is a line Patriot unit in the Continental United States. A CONUS TDA unit could be a Drill Sergeant or Recruiting assignment, for example, in the Continental United States. Any assignment Outside the Continental United States is considered an OCONUS assignment for the purposes of our model.

The differences in the types of locations are the criteria personnel managers use to determine a soldiers eligibility to move and the type of move generated from that type of location. Movement out of a OCONUS or a CONUS TDA assignment is based on a soldier reaching a certain TOS. For an assignment to Germany, for example, when a soldier has been on station for 30 months, he or she will be looked at for reassignment.

Assignments in Korea are normally 12 months and CONUS TDA assignments are 24-36 months in length. The difference here is that a move from a CONUS TDA assignment can be an "Operational move". Our model counts all moves to or from an OCONUS assignment as either Rotational or as an Accession or Separation move. Soldiers in a CONUS TOE unit are only moved when there is a requirement at another location, and usually only after 24 months TOS. If there is no requirement, a soldier remains at his or her current location.

3.3.2 Processing Blocks

We further break down the processing at the different locations into three blocks: Administrative, Orders, and Movement. In the Administrative block we account for promotions, demotions and attritions. In the Orders block, soldiers are "given" orders for reassignment to another location, or unit. Finally, in the Movement block, a soldier "PCSs" to his or her new duty location. We will discuss the actual coding of these blocks in a later section.

4.0 Characteristics of the Model

4.1 Locations

For our specific model, we defined eight separate locations a soldier can "move" to or from. Soldiers can be assigned to the Advanced Individual Training location at Fort Bliss, a CONUS TOE (Table of Organization and Equipment) unit at either Fort Bliss, Fort Hood, or Fort Polk. They can also be assigned to an OCONUS unit in Germany or in Korea. Finally they can be assigned to a CONUS TDA unit. We aggregated all the

CONUS TDA (Table of Distribution and Allowances) assignments into one unit. We did this because our client was not interested in the cost of each move, simply the number of moves, by type. This allowed us to consider a move from the Wisconsin National Guard to Fort Bliss the same as a move from Washington, DC to Fort Polk, for example.

4.2 Entities

There is only one type of entity in our model: a soldier. We model each 16T individually. This requires more memory than aggregating the entities but allow for greater resolution as to the individual characteristics of a soldier. We account for the various characteristics of the soldiers by assigning different attributes to each entity.

4.3 Attributes

We defined four attributes which we tracked for each individual soldier:

4.3.1 Skill Level (SL)

This is self-explanatory. As a soldier in this model is promoted to skill level 5 (E8), he or she leaves the system. This is because that soldier would then become a 16Z. This is an entirely different MOS and is managed separately.

4.3.2 Time on Station (TOS)

This is the number of months a soldier is at a particular location. This is incremented every month and is the key indicator to determine *when* a soldier is eligible for reassignment.

4.3.3 Permanent Change of Station Code (PCS_CODE)

This a code which is assigned to an entity which essentially, "put a soldier on orders" to another location. The table below shows the meaning of the various PCS Codes:

PCS Code	Assignment Instructions
1	Leave the system (leave the Army)
2	Fort Bliss, Texas
3	Germany
4	Fort Hood, Texas
5	Fort Polk, Louisiana
6	Korea
7	CONUS TDA
-1	Currently not on assignment instructions

Table 1: PCS Codes

4.3.4 Time Until Movement (TTM)

This is a value given to an entity to signify the number of months until that entity proceeds to its new assignment after being "put on" assignment instructions. Prior to placing a soldier on orders, this number is negative. After a soldier is put on orders, he or she is assigned a positive number based on the need of the gaining unit. We discuss the actual assignment algorithm later. An entity's TTM is decremented every time period (month). When it becomes zero, the entity moves to the new "unit".

4.4 Variables

Most of the variables we use in this model are defined to calculate which unit a solider ready for PCS should be assigned to. An assignment officer generally reassigns a PCS eligible soldier to the unit with the greatest "need". This is based on a priority fill plan. Under this plan, different units are designated to have a higher priority. Personnel managers attempt to keep these units at high fill percentages. To replicate the assignment process, we defined a great number of variables in this model. We defined each of the variables explained below for each skill level and location. For example, the actual number of skill level 1 soldiers at Fort Bliss is defined as Bliss_act1.

4.4.1 Target

Different units have different "fill priorities". Some units, such as rapid deployment units, have a high fill priority. Personnel managers try and maintain a fill level of over 100% for these units. This way, even with normal fluctuations in personnel levels, a high priority unit will never drop below 100% fill. There are "bill payers" for this over-manning. These are the low priority units. Personnel managers are willing to let these units fill levels drop below 100%, perhaps as low as 95%. To account for these different priority units, we used a "target" value instead of simply a unit's authorized strength. For example, if a unit is a high priority unit and has an authorized strength of 200 personnel, the target value would be 204 (200*102%). Similarly, if a low priority unit has an authorized strength of 100, its "target" could be 98 or even 95 depending on the priority. The usefulness of this definition will be more transparent when we discuss the "need" variable below.

4.4.2 Actual

The variable is simply the actual number of soldiers, by skill level, at a given installation. This variable is incremented when an entity arrives at a location.

4.4.3 Gain

This is the number of soldiers "on orders" for a given installation, by skill level. When an entity is assigned a PCS Code, thereby designating a location to where that entity will move, this variable is incremented. When the entity is considered and arrival (thus incrementing the actual counter at a location) this variable is decremented.

4.4.4 Loss

This is obviously just the opposite of a gain variable. When an entity is assigned a PCS Code, this variable is incremented as a loss at that location. Upon arrival a the new location, this variable is decremented.

4.4.5 Need

This is where all the variables come together to calculate a unit's "need" for a new soldier. The formula we use is simple: NEED = TARGET - ACTUAL - GAIN + LOSS. We calculate this for each skill level and each location. For example, the calculation of Fort Hood's need of skill level 1 soldiers is: HOOD_NEED1 = HOOD_TGT1 - HOOD_ACT1 - HOOD_GAIN1 + HOOD_LOSS1. Based on this calculation, a high priority unit would have a higher need even when its actual strength (measured against authorized) is higher than that of a low priority unit.

4.5 Processing

As we stated before, there are three "blocks" of processing in each location, Administrative, Orders, and Movement. When an entity arrives at a given location, the processing first increments the "actual" variable and decrements the "gain" variable for that location. A new "overall actual" value for that location is then calculated. We depict this processing in Table 2, below.

```
IF SL=1 THEN
        BEGIN
                INC BLISS_ACT1
        DEC BLISS_GAIN1
        END
IF SL=2 THEN
        BEGIN
                INC BLISS_ACT2
                DEC BLISS_GAIN2
        END
IF SL=3 THEN
        BEGIN
                INC BLISS_ACT3
                DEC BLISS_GAIN3
        END
IF SL=4 THEN
        BEGIN
                INC BLISS_ACT4
                DEC BLISS_GAIN4
        END

BLISS_ACT = (BLISS_ACT1+BLISS_ACT2+BLISS_ACT3+BLISS_ACT4)
```

Table 2: Opening Processing Code for Fort Bliss Location

Once we have updated the variables, the entity enters a DO/UNTIL Loop. The entity remains in this loop until it is assigned a PCS_CODE of 1 (leave the system, or ETS) or its TTM = 0.

4.5.1 Administrative Processing

During any given month, a soldier either attrites, is promoted, demoted, or is continued at the same skill level another month. The rate at which soldiers in a given skill level and MOS complete one of these actions in a given month varies wildly depending on numerous factors. To simplify our calculations, we used parameters used in the Enlisted Loss Inventory Model, or ELIM. ELIM is the official model that the Army personnel community uses to manage the enlisted force. This model calculates a thirty-six month weighted average to compute rates for attritions, promotions and demotions by skill level and MOS. These values are given as deterministic parameters. We replicate the randomness of this process by generating a random number and comparing it to a "yardstick". This yardstick varies from 0 to 1. Within this yardstick are "bands" for attrition, promotion, demotion, and continuation as seen in the figure below:

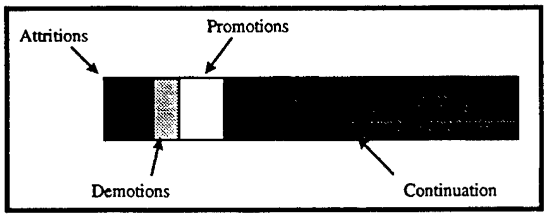

Figure 2: Personnel Action Bar

The model then compares the random value generated against this yardstick and depending on where it "lands", the appropriate personnel action is taken on the entity. For a promotion, the skill level is incremented and it is decremented for a demotion[1]. In the case of an attrition, the entity is assigned a PCS_CODE = 1 and then leaves the system. This process of generating a new random number and then comparing a random number against a yardstick every time increment could allow an entity to be promoted one month, promoted the next month, demoted the next month, promoted again the net and then finally attrited (or any combination). This may not seem realistic for a given soldier. We are not concerned with this because though we are tracking individual entities, we are only interested in information from the aggregate. The individual promotions, demotions and attritions are unimportant to our analysis. We are merely interested in the fact that in a given installation, for a given month, for a given skill level and MOS, "roughly" a certain percentage will fall into various administrative actions. Refer to Annex B for the processing code in this block.

[1]Skill level 1 soldiers can only attrite, be promoted and continued.

4.5.2 Orders Assignment Processing

Embedded with the Administrative Processing code is the Orders Assignment Processing. When a soldier reaches a certain Time on Station (TOS) "threshold", an assignment manager begins to look at that soldier for reassignment. The trigger mechanism is different depending on the type of assignment or the location of the assignment. For example, if a soldier is assigned to Germany, an assignment manager will look for that soldier to redeploy based solely on that soldier's TOS, regardless of the readiness of his or her current unit or other units. The same can be said for most TDA assignments. These assignments last two or three years generally depending on the type of assignment. When those two or three years are over, the soldier moves to a new location. Again, the reassignment trigger is TOS.

For a soldier in a CONUS TOE unit, however, there is more involved. These soldiers will generally stay on station at least 24 months prior to being allowed to be reassigned elsewhere. Once this soldier attains that TOS threshold, he or she can be reassigned if there is a greater need elsewhere. In the case of moving the soldier to another CONUS TOE unit (thus generating an Operational move), this need must be much greater.

We attempted to account for these realities in our model. Once an entity reaches a given TOS threshold, it enters the orders processing block. Once in this block, the need for a given location is calculated for the given skill level.[2] We do a pair-wise comparison and calculate the MAX NEED. This is the location with the greatest need for this skill

[2] Because promotions are instantaneous, there is no "promotable" status. This means that we do not assign an E5(P) to an E6 position. We also do not allow a skill level one soldier to move to skill level 2 job,

level entity. For a OCONUS or a TDA location, the soldier is then assigned a PCS_CODE corresponding to the location with the greatest need. In the case of a CONUS TOE unit, the need at the MAX NEED location must be significantly greater than that at the current location. If it is just slightly greater, there is not need to move the soldier immediately. In this case, the entity is assigned a PCS_CODE of -1 (not on orders) and continues back to the administrative block for processing the next month with a higher TOS.

After an entity is assigned a PCS_CODE >1, the process then calculates the Time to Move (TTM) for the entity. AS the MAX NEED increases, the TTM decreases in a step-wise manner. If the MAX NEED is high, the entity will move in two months. If the need is low, the entity will move in six or more months.

In our simulation, we also must account for a soldier electing to remain at an overseas location for another tour. This is termed a Consecutive Overseas Tour (COT). For example, a soldier in Germany can request to have another 36 month tour immediately after his initial 36 month tour is completed. A soldier can request waivers to reduce the length of either the initial or subsequent tour. With these waivers, the overall tour length could vary from 48 months to 72 months. Under another program, soldiers can also extend their tours beyond the regular 36 months for months ranging from 2 to 24.[3] We account for this in the model by again comparing the random number generated in the administrative block with a COT rate. [Black, Jul 96] Once an entity "accepts" a

---

regardless of the need. If this feature is desired, then we recommend calculating the MAX NEED to include one lower skill level and one higher skill level and assigning an entity based on this new calculation.

[3] Overseas tours in Korea work much the same except that the time lengths are different. A soldier can also move from Germany to Korea on a COT and vice versa. E do not account for this but the model can be easily modified to do so.

COT, his TOS is adjusted based on a distribution to reflect the variability in the length of the extension or new tour length.

After the entity is assigned a PCS_CODE and a TTM, it returns to the administrative block and continues to be compared to be attrited, promoted, demoted or continued. The TTM is decremented every month in this block. Upon reaching zero, the soldier moves to the movement processing.

4.5.3 Movement Processing

We liken the movement of entities around the simulation to a monorail system. After leaving the location processing, the simulation analyzes the PCS_CODE attribute of the entity. The entity is then moved to the location alluded to by this attribute. Prior to movement, the PCS_CODE is reset to -1, the TTM is reset to -1, The TOS is reset to 0, and the "actual" and "loss" variables for that skill level at the location are decremented. If the movement is between two CONUS TOE locations or a CONUS TOE and a CONUS TDA location, the move is an operational move. In this instance, the OPER_MOVE variable is incremented. Finally, the entity is told to wait either one (for a move between CONUS units) or two (for a move overseas) time increments to make the move. The table below depicts an example of the movement processing logic. In this case, this is a move between Fort Polk and Fort Hood.

```
INC OPER_MOVE
IF SL = 1 THEN
        BEGIN
                DEC POLK_ACT1
                INC POLK_LOSS1
        END
IF SL = 2 THEN
        BEGIN
                DEC POLK_ACT2
                INC POLK_LOSS2
        END
IF SL =3 THEN
```

```
          BEGIN
                    DEC POLK_ACT3
                    INC POLK_LOSS3
          END
IF SL = 4 THEN
          BEGIN
                    DEC POLK_ACT4
                    INC POLK_ACT4
          END
TOS = 0
TTM = -1
PCS_CODE = -1
MOVE FOR 1
```

Table 3: Movement Processing Code for move from Fort Polk to Fort Hood

4.6 Arrivals

Entities arrive to the system in only one location: Fort Bliss AIT. We do not manage the arrivals or the processing at this location. This model is not interested in recruiting rates or retention rates at Basic or AIT. Each entity which arrives at the AIT location, moves to the unit with the greatest need for skill level one soldiers immediately. We establish the arrivals at this location to correspond with the AIT graduation rates we received from the ELIM output [Hersh, 1995]. These rates can be modified once the interface of the model has been established to fully validate the model.

5.0 Verification of the Model

We have run the model a number of times under various initial conditions. Promotions occur in a realistic fashion as shown in the figure below. In this figure, the lines represent the actual number of soldiers, by skill level at Fort Bliss at a given time. Note that the time scale along the x-axis is in hours. This simulation package does not allow "months" as a time increment. We used minutes to represent months. Therefore, one hour of simulation time would equal 60 months of "real" time. As you can see, soldiers are being attriting and/or moving out of the location. This is a graph taken directly from the ProModel® statistics package.

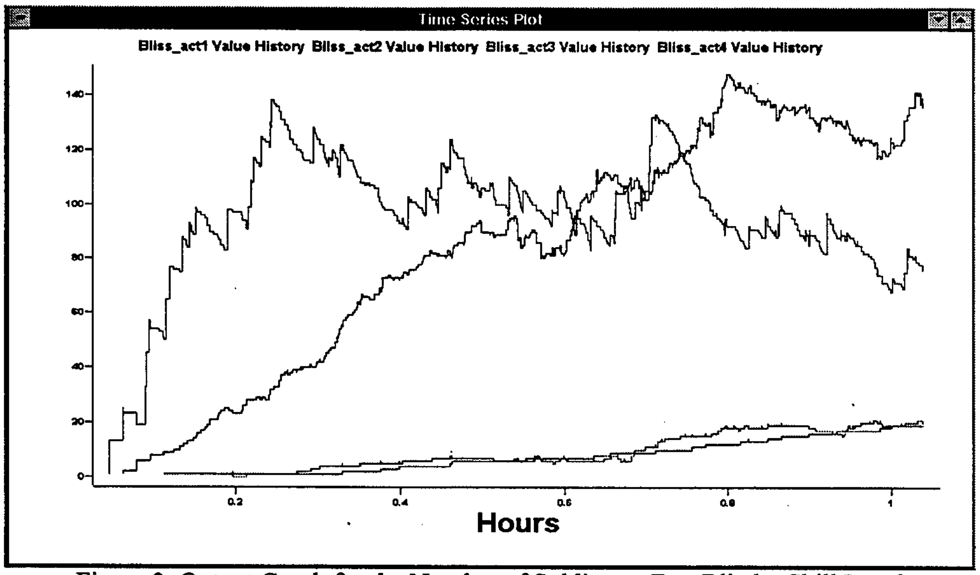

Figure 3: Output Graph for the Number of Soldiers at Fort Blissby Skill Level

Figure 4, below, shows the number of soldiers at the different locations. Note that the system stabilizes after about 30 months. This verifies that the model is operating as we intended and should continue to do so when we modify the initial conditions.

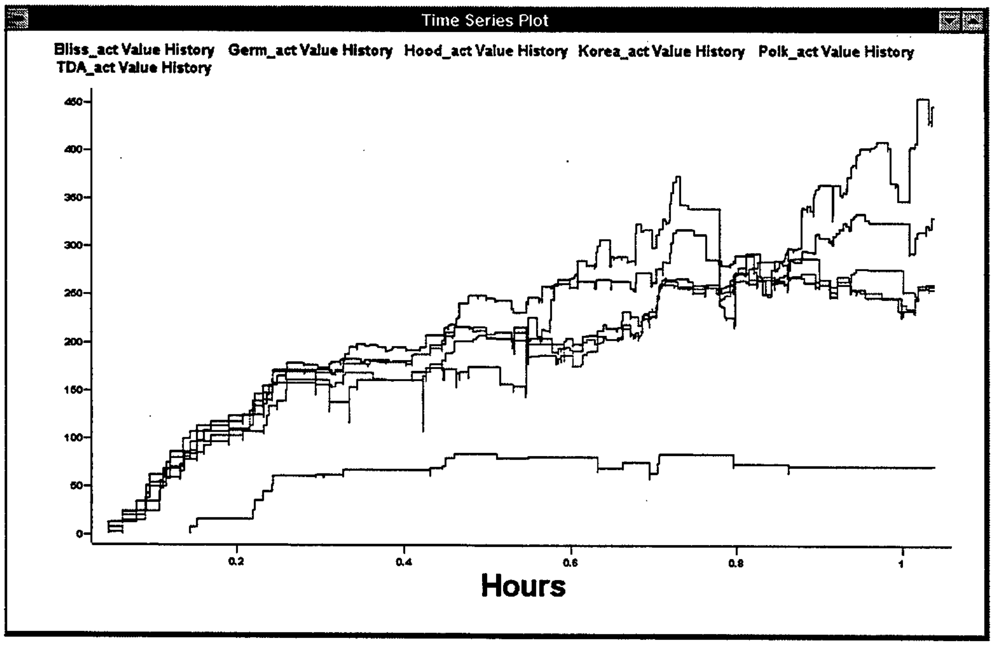

Figure 4: Output Graph for Overall Number of Soldiers at all Locations

6.0 Validation and Future Development

In order to fully validate the model we must begin with realistic data on soldiers, their current locations and other personal data. A front-end interface with the Enlisted Master File must be developed to do this. This should be an easy step for an individual with access to the database and programming skills. The database must be queried and sorted into 16T soldiers. The programmer must then modify this data so that it is in the format required by ProModel. For example, for each soldier with a Fort Bliss Unit Identifier Code (UIC), assign the entity with a 2 in the Location column.

Once this interface is established, we must modify the rates, the TOS thresholds and the TTM values to obtain realistic annual operational move numbers. We should also see the correct number of promotions, demotions and attritions. Since the model seems to be operating correctly, the only modifications should be in these rates. Once fully validated, the model can be used in analysis, but only for 16T MOS.

In order to use this simulation for evaluation of PCS policies for the entire Army, the model should be expanded to include all MOSs. On the surface this seems an easy task. To expand to different MOSs, simply assign another attribute to each entity delineating that entities MOS. The problem becomes computational. The sheer number of soldiers in the Army requires that some aggregation be done in order to expand this model. The question becomes: what do you aggregate? As you aggregate, you lose some resolution in the model. This will be a problem. The use of ProModel exacerbates the problem. ProModel is an easy-to-use, visual simulation package. The problem is that it requires a great deal of memory to use the animation and the other features of the package. A user might be able to do less aggregation with a different simulation package.

7.0 Use of the Model in Analysis

Once the EMF interface is in place and the model is fully validated, it should be a great analysis tool for personnel managers. This simulation was designed to assist a personnel manager, or analyst, in determining the effect of changing TOS policies on the number of moves and level of readiness of a unit[4]. After modifying the TOS thresholds to reflect the policy recommendation, a user simply runs the simulation. The simulation will track the number of operational moves required under this new policy. The user can then compare the number of moves under the current policy with the number under the proposed policy.

The user can also view the impact of the policy on readiness of a unit. After running the simulation, the user views the statistics from the run. By analyzing the graphs, (which is extremely simple to create in ProModel®), a user can see where the drops are in units readiness, how deep these drops are and how long the system will take to recover completely.

For example, suppose a decision maker recommends a policy which increases the tour lengths in Germany from 36 to 42 months. An analyst would first run the model under the current 36 month TOS policy and obtain the baseline results. Then the analyst would re-run the model after changing the TOS threshold for Germany. The analyst would simply have to compare the results of the two runs to determine the impacts of such a policy. One would expect that this would initially increase the number of operational moves because soldiers would not return from overseas on their "projected" rotation cycle. There may also be a noticeable drop in readiness of CONUS units. The

[4] We measure readiness simply as the overall number of soldiers in a unit versus the authorized. We do not account for non-deployables or other measures of personnel readiness.

analyst would easily show a decision maker (using the graphs in the ProModel® statistics package) the effects of this policy after one, two or five years.

8.0 Conclusions

In this proof of principle we provide a worthwhile analysis tool for analyzing PCS policies for 16Ts. Considerable effort will make it a worthwhile analysis tool for the entire Army's personnel picture. Due to the modular design approach, this job will be considerably easier. The size of the model will be a problem but should not stall efforts to expand this model. In this time of ever-reducing budgets and personnel drawdowns, personnel managers need better analysis tools to provide meaningful answers to decision makers. This is a first step in that direction. It should not be the last. This project, or a similar effort, should be expanded to include all Army MOSs. We can no longer afford to analyze policies *after* we implement them.

Annex A: References

Adelsberger, Bernard, *Troop budget needs fixing*, Army Times, Dec 4, 95

Army Training Requirements and Resources System (ATRRS) Selected Actual Summaries run, 16T FY94-96 Actual Grads, 27 Mar 96.

Bonder, Seth, *Impact of the New Global Environment on US National Security Planning - Challenges to the OR Community*, International Transactions in Operational Research Vol. 1, No. 1, Mar 94.

Costello, John, Major General, State of the Air Defense Artillery, slides from briefing to faculty, USMA, 5 Mar 96.

General Research Corporation (GRC), *Military Occupational Specialty Level System (MOSLS) Advanced Training Course*, course packet, 13 Mar 96.

Hersh, Doug, Office of the Deputy Chief of Staff, Personnel, unpublished message on Army Enlisted End-Strength data, 28 Sep 95.

Hix, Mike, *Assignment Stability: An In-Progress Review*, RAND briefing, 28 September 1995.

Interview with General Dennis Reimer, Army Chief of Staff on Nov 21, 1995, Army Times, Dec 4, 95.

Kwinn, B.T., Muhammad, A., *Continuous Time Simulation Approach to Developing US Army Enlisted Personnel Policy*, work in progress, submitted for 1996 MORS Conference.

McGinnis, M. L., Kays, J. L., and Slaten, Pamela, *Computer Simulation of US Army Officer Professional Development*, Winter Simulation Conference Proceedings, Dec 1994.

McGinnis, M. L., and Fernandez, E., *Initial Entry Training for the United States Army: Optimal and Heuristic Scheduling Procedures*, IEEE Decision and Control Proceedings, 1995.

Operations Management Division, EPMD, PERSCOM, *Assignment Nomination Subsystem*, Briefing Packet, 10 Oct 95.

Office of the Deputy Chief of Staff, Personnel, MOSEL run, 20 Oct 95.

Office of Enlisted Manpower Analysis, Department of Social Studies, United States Military Academy, *PCS Study Policy Execution*, Briefing Packet, Nov 95.

Park, Chan S., *Contemporary Engineering Economics*, Third Edition, Addison-Weslsley Publishing Company, New York, May 1994.

Stover, E., Tucker, B. and Zopelis, J., briefing to MG James Costello, *Enlisted PCS Simulation Model*, 5 Mar 96.

Thorpe, Robert, *Enlisted Distribution Target Model*, Information Paper, Planning and Analysis Branch, EPMD, PERSCOM, Sep 1, 95.

Annex B: Processing Computer Code

Fort Bliss

```
IF SL=1 THEN
      BEGIN
        INC BLISS_ACT1
        DEC BLISS_GAIN1
      END
IF SL=2 THEN
      BEGIN
        INC BLISS_ACT2
        DEC BLISS_GAIN2
      END
IF SL=3 THEN
      BEGIN
        INC BLISS_ACT3
        DEC BLISS_GAIN3
      END
IF SL=4 THEN
      BEGIN
        INC BLISS_ACT4
        DEC BLISS_GAIN4
      END

BLISS_ACT = (BLISS_ACT1+BLISS_ACT2+BLISS_ACT3+BLISS_ACT4)

DO

      BEGIN  //Bliss block

        WAIT 1
        VALUE = RAND(1)
        INC TOS
        DEC TTM

        IF SL=1 THEN

          BEGIN  //SL1 Admin block

            IF VALUE < 0.029435 THEN
              BEGIN
                    PCS_CODE = 1
              END
            IF VALUE > 0.029435 AND VALUE < 0.076849 THEN
              BEGIN
                    INC SL
                    DEC BLISS_ACT1
                    INC BLISS_ACT2
              END

            IF TOS > 24 AND PCS_CODE < 1 THEN
```

```
BEGIN //Orders block

        NEED1 = -1000
        BLISS_NEED1=(BLISS_TGT1-BLISS_ACT1-BLISS_GAIN1+BLISS_LOSS1)
        GERM_NEED1=(GERM_TGT1-GERM_ACT1-GERM_GAIN1+GERM_LOSS1)
        POLK_NEED1=(POLK_TGT1-POLK_ACT1-POLK_GAIN1+POLK_LOSS1)
        HOOD_NEED1=(HOOD_TGT1-HOOD_ACT1-HOOD_GAIN1+HOOD_LOSS1)
        KOREA_NEED1=(KOREA_TGT1-KOREA_ACT1-KOREA_GAIN1+KOREA_LOSS1)
        TDA_NEED1=(TDA_TGT1-TDA_ACT1-TDA_GAIN1+TDA_LOSS1)

        IF BLISS_NEED1 > NEED1 THEN
                BEGIN
                        PCS_CODE=-1
                        NEED1=BLISS_NEED1
                END
        IF GERM_NEED1 > NEED1 THEN
                BEGIN
                        PCS_CODE=3
                        NEED1=GERM_NEED1
                END
        IF POLK_NEED1 > NEED1 THEN
                BEGIN
                        PCS_CODE=4
                        NEED1=POLK_NEED1
                END
        IF HOOD_NEED1 > NEED1 THEN
                BEGIN
                        PCS_CODE=5
                        NEED1=HOOD_NEED1
                END
        IF KOREA_NEED1 > NEED1 THEN
                BEGIN
                        PCS_CODE=6
                        NEED1=KOREA_NEED1
                END
        IF TDA_NEED1 > NEED1 THEN
                BEGIN
                        PCS_CODE=7
                        NEED1=TDA_NEED1
                END

        IF NEED1 > 5 THEN TTM =6
        IF NEED1 > 10 THEN TTM =4
        IF NEED1 > 15 THEN TTM =2
        IF NEED1 < 5 THEN
                BEGIN
                        TTM=-1
                        PCS_CODE=-1
                END

        IF PCS_CODE = 3 THEN INC GERM_GAIN1
        IF PCS_CODE = 4 THEN INC POLK_GAIN1
        IF PCS_CODE = 5 THEN INC HOOD_GAIN1
        IF PCS_CODE = 6 THEN INC KOREA_GAIN1
```

```
            IF PCS_CODE = 7 THEN INC TDA_GAIN1

            IF PCS_CODE > 0 THEN INC BLISS_LOSS1

        END  //Orders block

    END //SL1 Admin block

  IF SL = 2 THEN

    BEGIN  //SL2 Admin block

      IF VALUE < 0.014132 THEN
        BEGIN
            PCS_CODE = 1
        END
      IF VALUE > 0.014132 AND VALUE < 0.0272651 THEN
        BEGIN
            INC SL
            DEC BLISS_ACT2
            INC BLISS_ACT3
        END
      IF VALUE > 0.0272651 AND VALUE < 0.0145146 THEN
        BEGIN
            DEC SL
            DEC BLISS_ACT2
            INC BLISS_ACT1
        END

      IF TOS > 24 AND PCS_CODE < 1 THEN

        BEGIN  //Orders block

            NEED2 = -1000
            BLISS_NEED2=(BLISS_TGT2-BLISS_ACT2-BLISS_GAIN2+BLISS_LOSS2)
            GERM_NEED2=(GERM_TGT2-GERM_ACT2-GERM_GAIN2+GERM_LOSS2)
            POLK_NEED2=(POLK_TGT2-POLK_ACT2-POLK_GAIN2+POLK_LOSS2)
            HOOD_NEED2=(HOOD_TGT2-HOOD_ACT2-HOOD_GAIN2+HOOD_LOSS2)
            KOREA_NEED2=(KOREA_TGT2-KOREA_ACT2-KOREA_GAIN2+KOREA_LOSS2)
            TDA_NEED2=(TDA_TGT2-TDA_ACT2-TDA_GAIN2+TDA_LOSS2)

            IF BLISS_NEED2 > NEED2 THEN
                BEGIN
                    PCS_CODE=-1
                    NEED2=BLISS_NEED2
                END
            IF GERM_NEED2 > NEED2 THEN
                BEGIN
                    PCS_CODE=3
                    NEED2=GERM_NEED2
                END
            IF POLK_NEED2 > NEED2 THEN
                BEGIN
                    PCS_CODE=4
                    NEED2=POLK_NEED2
                END
            IF HOOD_NEED2 > NEED2 THEN
                BEGIN
                    PCS_CODE=5
                    NEED2=HOOD_NEED2
                END
            IF KOREA_NEED2 > NEED2 THEN
                BEGIN
                    PCS_CODE=6
                    NEED2=KOREA_NEED2
                END
            IF TDA_NEED2 > NEED2 THEN
                BEGIN
                    PCS_CODE=7
                    NEED2=TDA_NEED2
                END

            IF NEED2 > 5 THEN TTM =6
            IF NEED2 > 10 THEN TTM =4
            IF NEED2 > 15 THEN TTM =2
            IF NEED2 < 5 THEN
                BEGIN
                    TTM=-1
                    PCS_CODE=-1
                END

            IF PCS_CODE = 3 THEN INC GERM_GAIN2
            IF PCS_CODE = 4 THEN INC POLK_GAIN2
            IF PCS_CODE = 5 THEN INC HOOD_GAIN2
            IF PCS_CODE = 6 THEN INC KOREA_GAIN2
            IF PCS_CODE = 7 THEN INC TDA_GAIN2

            IF PCS_CODE > 0 THEN INC BLISS_LOSS2

      END  //Orders block

  END  //SL2 Admin block

IF SL = 3 THEN

  BEGIN  //SL3 Admin block

    IF VALUE < 0.0071226 THEN
      BEGIN
            PCS_CODE = 1
      END
    IF VALUE > 0.0071226 AND VALUE < 0.0173445 THEN
      BEGIN
            INC SL
            DEC BLISS_ACT3
            INC BLISS_ACT4
      END
    IF VALUE > 0.0173445 AND VALUE < 0.0198757 THEN
      BEGIN
            DEC SL
      DEC BLISS_ACT3
      INC BLISS_ACT2
   END

IF TOS > 24 AND PCS_CODE < 1 THEN

   BEGIN  //SL3 Orders block

      NEED3 = -1000
      BLISS_NEED3=(BLISS_TGT3-BLISS_ACT3-BLISS_GAIN3+BLISS_LOSS3)
      GERM_NEED3=(GERM_TGT3-GERM_ACT3-GERM_GAIN3+GERM_LOSS3)
      POLK_NEED3=(POLK_TGT3-POLK_ACT3-POLK_GAIN3+POLK_LOSS3)
      HOOD_NEED3=(HOOD_TGT3-HOOD_ACT3-HOOD_GAIN3+HOOD_LOSS3)
      KOREA_NEED3=(KOREA_TGT3-KOREA_ACT3-KOREA_GAIN3+KOREA_LOSS3)
      TDA_NEED3=(TDA_TGT3-TDA_ACT3-TDA_GAIN3+TDA_LOSS3)

      IF BLISS_NEED3 > NEED3 THEN
          BEGIN
              PCS_CODE=-1
              NEED3=BLISS_NEED3
          END
      IF GERM_NEED3 > NEED3 THEN
          BEGIN
              PCS_CODE=3
              NEED3=GERM_NEED3
          END
      IF POLK_NEED3 > NEED3 THEN
          BEGIN
              PCS_CODE=4
              NEED3=POLK_NEED3
          END
      IF HOOD_NEED3 > NEED3 THEN
          BEGIN
              PCS_CODE=5
              NEED3=HOOD_NEED3
          END
      IF KOREA_NEED3 > NEED3 THEN
          BEGIN
              PCS_CODE=6
              NEED3=KOREA_NEED3
          END
      IF TDA_NEED3 > NEED3 THEN
          BEGIN
              PCS_CODE=7
              NEED3=TDA_NEED3
          END

      IF NEED3 > 5 THEN TTM =6
      IF NEED3 > 10 THEN TTM =4
      IF NEED3 > 15 THEN TTM =2
      IF NEED3 < 5 THEN
          BEGIN
              TTM=-1
              PCS_CODE=-1
          END
```

```
            IF PCS_CODE = 3 THEN INC GERM_GAIN3
            IF PCS_CODE = 4 THEN INC POLK_GAIN3
            IF PCS_CODE = 5 THEN INC HOOD_GAIN3
            IF PCS_CODE = 6 THEN INC KOREA_GAIN3
            IF PCS_CODE = 7 THEN INC TDA_GAIN3

            IF PCS_CODE > 0 THEN INC BLISS_LOSS3

      END  //SL3 Orders block

  END  //SL3 Admin block

IF SL = 4 THEN

  BEGIN  //SL4 Admin block

    IF VALUE < 0.0129963 THEN
      BEGIN
            PCS_CODE = 1
      END
    IF VALUE > 0.0129963 AND VALUE < 0.0208022 THEN
      BEGIN
            INC SL
            PCS_CODE = 1
      END
    IF VALUE > 0.0208022 AND VALUE < 0.0211931 THEN
      BEGIN
            DEC SL
            DEC BLISS_ACT4
            INC BLISS_ACT3
      END

    IF TOS > 24 AND PCS_CODE < 1 THEN

      BEGIN  //SL4 Orders block

            NEED4 = -1000
            BLISS_NEED4=(BLISS_TGT4-BLISS_ACT4-BLISS_GAIN4+BLISS_LOSS4)
            GERM_NEED4=(GERM_TGT4-GERM_ACT4-GERM_GAIN4+GERM_LOSS4)
            POLK_NEED4=(POLK_TGT4-POLK_ACT4-POLK_GAIN4+POLK_LOSS4)
            HOOD_NEED4=(HOOD_TGT4-HOOD_ACT4-HOOD_GAIN4+HOOD_LOSS4)
            KOREA_NEED4=(KOREA_TGT4-KOREA_ACT4-KOREA_GAIN4+KOREA_LOSS4)
            TDA_NEED4=(TDA_TGT4-TDA_ACT4-TDA_GAIN4+TDA_LOSS4)

            IF BLISS_NEED4 > NEED4 THEN
                    BEGIN
                            PCS_CODE=-1
                            NEED4=BLISS_NEED4
                    END
            IF GERM_NEED4 > NEED4 THEN
                    BEGIN
                            PCS_CODE=3
                            NEED4=GERM_NEED4
                    END
```

```
                IF POLK_NEED4 > NEED4 THEN
                    BEGIN
                        PCS_CODE=4
                        NEED4=POLK_NEED4
                    END
                IF HOOD_NEED4 > NEED4 THEN
                    BEGIN
                        PCS_CODE=5
                        NEED4=HOOD_NEED4
                    END
                IF KOREA_NEED4 > NEED4 THEN
                    BEGIN
                        PCS_CODE=6
                        NEED4=KOREA_NEED4
                    END
                IF TDA_NEED4 > NEED4 THEN
                    BEGIN
                        PCS_CODE=7
                        NEED4=TDA_NEED4
                    END

                IF NEED4 > 5 THEN TTM =6
                IF NEED4 > 10 THEN TTM =4
                IF NEED4 > 15 THEN TTM =2
                IF NEED4 < 5 THEN
                    BEGIN
                        TTM=-1
                        PCS_CODE=-1
                    END

                IF PCS_CODE = 3 THEN INC GERM_GAIN4
                IF PCS_CODE = 4 THEN INC POLK_GAIN4
                IF PCS_CODE = 5 THEN INC HOOD_GAIN4
                IF PCS_CODE = 6 THEN INC KOREA_GAIN4
                IF PCS_CODE = 7 THEN INC TDA_GAIN4

                IF PCS_CODE > 0 THEN INC BLISS_LOSS4

            END //SL4 Orders block

        END //SL4 Admin block

    END //Bliss block

UNTIL TTM = 0 OR PCS_CODE = 1
```

Germany

```
IF SL=1 THEN
      BEGIN
        INC GERM_ACT1
        DEC GERM_GAIN1
      END
IF SL=2 THEN
      BEGIN
        INC GERM_ACT2
        DEC GERM_GAIN2
      END
IF SL=3 THEN
      BEGIN
        INC GERM_ACT3
        DEC GERM_GAIN3
      END
IF SL=4 THEN
      BEGIN
        INC GERM_ACT4
        DEC GERM_GAIN4
      END

GERM_ACT = (GERM_ACT1+GERM_ACT2+GERM_ACT3+GERM_ACT4)

DO

      BEGIN  //Germany block

        WAIT 1
        VALUE = RAND(1)
        INC TOS
        DEC TTM

        IF SL=1 THEN

          BEGIN  //SL1 Admin block

            IF VALUE < 0.029435 THEN
              BEGIN
                    PCS_CODE = 1
              END
            IF VALUE > 0.029435 AND VALUE < 0.076849 THEN
              BEGIN
                    INC SL
                    DEC GERM_ACT1
                    INC GERM_ACT2
              END

            IF TOS > 30 AND PCS_CODE < 1 THEN

              BEGIN  //SL1 Orders block

                    NEED1 = -1000
```

```
BLISS_NEED1=(BLISS_TGT1-BLISS_ACT1-BLISS_GAIN1+BLISS_LOSS1)
GERM_NEED1=(GERM_TGT1-GERM_ACT1-GERM_GAIN1+GERM_LOSS1)
POLK_NEED1=(POLK_TGT1-POLK_ACT1-POLK_GAIN1+POLK_LOSS1)
HOOD_NEED1=(HOOD_TGT1-HOOD_ACT1-HOOD_GAIN1+HOOD_LOSS1)
KOREA_NEED1=(KOREA_TGT1-KOREA_ACT1-KOREA_GAIN1+KOREA_LOSS1)
TDA_NEED1=(TDA_TGT1-TDA_ACT1-TDA_GAIN1+TDA_LOSS1)

IF BLISS_NEED1 > NEED1 THEN
        BEGIN
                PCS_CODE=2
                NEED1=BLISS_NEED1
        END
IF POLK_NEED1 > NEED1 THEN
        BEGIN
                PCS_CODE=4
                NEED1=POLK_NEED1
        END
IF HOOD_NEED1 > NEED1 THEN
        BEGIN
                PCS_CODE=5
                NEED1=HOOD_NEED1
        END
IF KOREA_NEED1 > NEED1 THEN
        BEGIN
                PCS_CODE=6
                NEED1=KOREA_NEED1
        END
IF TDA_NEED1 > NEED1 THEN
        BEGIN
                PCS_CODE=7
                NEED1=TDA_NEED1
        END

IF VALUE < 0.10 THEN
        BEGIN
                PCS_CODE = -1
                TTM=-1
                TOS = N(14,8)
        END

ELSE
        BEGIN
                INC GERM_LOSS1
                IF NEED1 < 0 THEN TTM = 6
                IF NEED1 > 0 THEN TTM = 5
                IF NEED1 > 5 THEN TTM =4
                IF NEED1 > 10 THEN TTM = 3
                IF NEED1 > 15 THEN TTM =2
        END

IF PCS_CODE = 2 THEN INC BLISS_GAIN1
IF PCS_CODE = 4 THEN INC POLK_GAIN1
IF PCS_CODE = 5 THEN INC HOOD_GAIN1
IF PCS_CODE = 6 THEN INC KOREA_GAIN1
IF PCS_CODE = 7 THEN INC TDA_GAIN1
```

```
        END  //SL1 Orders block

    END  //SL1 Admin block

  IF SL = 2 THEN

    BEGIN  //SL2 Admin block

      IF VALUE < 0.014132 THEN
        BEGIN
              PCS_CODE = 1
        END
      IF VALUE > 0.014132 AND VALUE < 0.0272651 THEN
        BEGIN
              INC SL
              INC GERM_ACT3
              DEC GERM_ACT2
        END
      IF VALUE > 0.0272651 AND VALUE < 0.0145146 THEN
        BEGIN
              DEC SL
              DEC GERM_ACT2
              INC GERM_ACT1
        END

      IF TOS > 30 AND PCS_CODE < 1 THEN

        BEGIN  //SL2 Orders block

              NEED2 = -1000
              BLISS_NEED2=(BLISS_TGT2-BLISS_ACT2-BLISS_GAIN2+BLISS_LOSS2)
              GERM_NEED2=(GERM_TGT2-GERM_ACT2-GERM_GAIN2+GERM_LOSS2)
              POLK_NEED2=(POLK_TGT2-POLK_ACT2-POLK_GAIN2+POLK_LOSS2)
              HOOD_NEED2=(HOOD_TGT2-HOOD_ACT2-HOOD_GAIN2+HOOD_LOSS2)
              KOREA_NEED2=(KOREA_TGT2-KOREA_ACT2-KOREA_GAIN2+KOREA_LOSS2)
              TDA_NEED2=(TDA_TGT2-TDA_ACT2-TDA_GAIN2+TDA_LOSS2)

              IF BLISS_NEED2 > NEED2 THEN
                      BEGIN
                              PCS_CODE=2
                              NEED2=BLISS_NEED2
                      END
              IF POLK_NEED2 > NEED2 THEN
                      BEGIN
                              PCS_CODE=4
                              NEED2=POLK_NEED2
                      END
              IF HOOD_NEED2 > NEED2 THEN
                      BEGIN
                              PCS_CODE=5
                              NEED2=HOOD_NEED2
                      END
              IF KOREA_NEED2 > NEED2 THEN
                      BEGIN
```

```
                        PCS_CODE=6
                        NEED2=KOREA_NEED2
                END
        IF TDA_NEED2 > NEED2 THEN
                BEGIN
                        PCS_CODE=7
                        NEED2=TDA_NEED2
                END

        IF VALUE < 0.10 THEN
                BEGIN
                         PCS_CODE = -1
                        TTM=-1
                        TOS = N(14,8)
                END

        ELSE
                BEGIN
                        INC GERM_LOSS2
                        IF NEED1 < 0 THEN TTM = 6
                        IF NEED1 > 0 THEN TTM = 5
                        IF NEED1 > 5 THEN TTM =4
                        IF NEED1 > 10 THEN TTM = 3
                        IF NEED1 > 15 THEN TTM =2
                END

        IF PCS_CODE = 2 THEN INC BLISS_GAIN2
        IF PCS_CODE = 4 THEN INC POLK_GAIN2
        IF PCS_CODE = 5 THEN INC HOOD_GAIN2
        IF PCS_CODE = 6 THEN INC KOREA_GAIN2
        IF PCS_CODE = 7 THEN INC TDA_GAIN2

   END  //SL2 Orders block

 END  //SL2 Admin block

IF SL = 3 THEN

 BEGIN  //SL3 Admin block

  IF VALUE < 0.0071226 THEN
   BEGIN
        PCS_CODE = 1
   END
  IF VALUE > 0.0071226 AND VALUE < 0.0173445 THEN
   BEGIN
        INC SL
        INC GERM_ACT4
        DEC GERM_ACT3
   END
  IF VALUE > 0.0173445 AND VALUE < 0.0198757 THEN
   BEGIN
        DEC SL
        INC GERM_ACT2
        DEC GERM_ACT3
   END

IF TOS > 30 AND PCS_CODE < 1 THEN

   BEGIN  //SL3 Orders block

         NEED3 = -1000
         BLISS_NEED3=(BLISS_TGT3-BLISS_ACT3-BLISS_GAIN3+BLISS_LOSS3)
         GERM_NEED3=(GERM_TGT3-GERM_ACT3-GERM_GAIN3+GERM_LOSS3)
         POLK_NEED3=(POLK_TGT3-POLK_ACT3-POLK_GAIN3+POLK_LOSS3)
         HOOD_NEED3=(HOOD_TGT3-HOOD_ACT3-HOOD_GAIN3+HOOD_LOSS3)
         KOREA_NEED3=(KOREA_TGT3-KOREA_ACT3-KOREA_GAIN3+KOREA_LOSS3)
         TDA_NEED3=(TDA_TGT3-TDA_ACT3-TDA_GAIN3+TDA_LOSS3)

         IF BLISS_NEED3 > NEED3 THEN
                 BEGIN
                         PCS_CODE=2
                         NEED3=BLISS_NEED3
                 END
         IF POLK_NEED3 > NEED3 THEN
                 BEGIN
                         PCS_CODE=4
                         NEED3=POLK_NEED3
                 END
         IF HOOD_NEED3 > NEED3 THEN
                 BEGIN
                         PCS_CODE=5
                         NEED3=HOOD_NEED3
                 END
         IF KOREA_NEED3 > NEED3 THEN
                 BEGIN
                         PCS_CODE=6
                         NEED3=KOREA_NEED3
                 END
         IF TDA_NEED3 > NEED3 THEN
                 BEGIN
                         PCS_CODE=7
                         NEED3=TDA_NEED3
                 END

         IF VALUE < 0.10 THEN
                 BEGIN
                          PCS_CODE = -1
                         TTM=-1
                         TOS = N(14,8)
                 END

         ELSE
                 BEGIN
                         INC GERM_LOSS3
                         IF NEED1 < 0 THEN TTM = 6
                         IF NEED1 > 0 THEN TTM = 5
                         IF NEED1 > 5 THEN TTM =4
                         IF NEED1 > 10 THEN TTM = 3
                         IF NEED1 > 15 THEN TTM =2
```

```
                END

            IF PCS_CODE = 2 THEN INC BLISS_GAIN3
            IF PCS_CODE = 4 THEN INC POLK_GAIN3
            IF PCS_CODE = 5 THEN INC HOOD_GAIN3
            IF PCS_CODE = 6 THEN INC KOREA_GAIN3
            IF PCS_CODE = 7 THEN INC TDA_GAIN3

        END  //SL3 Orders block

    END  //SL3 Admin block

IF SL = 4 THEN

    BEGIN  //SL4 Admin block

      IF VALUE < 0.0129963 THEN
        BEGIN
            PCS_CODE = 1
        END
      IF VALUE > 0.0129963 AND VALUE < 0.0208022 THEN
        BEGIN
            INC SL
            PCS_CODE = 1
        END
      IF VALUE > 0.0208022 AND VALUE < 0.0211931 THEN
        BEGIN
            DEC SL
            INC GERM_ACT3
            DEC GERM_ACT4
        END

      IF TOS > 30 AND PCS_CODE < 1 THEN

        BEGIN  //SL4 Orders block

            NEED4 = -1000
            BLISS_NEED4=(BLISS_TGT4-BLISS_ACT4+BLISS_GAIN4-BLISS_LOSS4)
            GERM_NEED4=(GERM_TGT4-GERM_ACT4+GERM_GAIN4-GERM_LOSS4)
            POLK_NEED4=(POLK_TGT4-POLK_ACT4+POLK_GAIN4-POLK_LOSS4)
            HOOD_NEED4=(HOOD_TGT4-HOOD_ACT4+HOOD_GAIN4-HOOD_LOSS4)
            KOREA_NEED4=(KOREA_TGT4-KOREA_ACT4+KOREA_GAIN4-KOREA_LOSS4)
            TDA_NEED4=(TDA_TGT4-TDA_ACT4+TDA_GAIN4-TDA_LOSS4)

            IF BLISS_NEED4 > NEED4 THEN
                BEGIN
                    PCS_CODE=2
                    NEED4=BLISS_NEED4
                END
            IF POLK_NEED4 > NEED4 THEN
                BEGIN
                    PCS_CODE=4
                    NEED4=POLK_NEED4
                END
            IF HOOD_NEED4 > NEED4 THEN
```

```
                BEGIN
                        PCS_CODE=5
                        NEED4=HOOD_NEED4
                END
        IF KOREA_NEED4 > NEED4 THEN
                BEGIN
                        PCS_CODE=6
                        NEED4=KOREA_NEED4
                END
        IF TDA_NEED4 > NEED4 THEN
                BEGIN
                        PCS_CODE=7
                        NEED4=TDA_NEED4
                END

        IF VALUE < 0.10 THEN
                BEGIN
                         PCS_CODE = -1
                        TTM=-1
                        TOS = N(14,8)
                END

        ELSE
                BEGIN
                        INC GERM_LOSS4
                        IF NEED1 < 0 THEN TTM = 6
                        IF NEED1 > 0 THEN TTM = 5
                        IF NEED1 > 5 THEN TTM =4
                        IF NEED1 > 10 THEN TTM = 3
                        IF NEED1 > 15 THEN TTM =2
                END

        IF PCS_CODE = 2 THEN INC BLISS_GAIN4
        IF PCS_CODE = 4 THEN INC POLK_GAIN4
        IF PCS_CODE = 5 THEN INC HOOD_GAIN4
        IF PCS_CODE = 6 THEN INC KOREA_GAIN4
        IF PCS_CODE = 7 THEN INC TDA_GAIN4

      END  //SL4 Orders block

    END  //SL4 Admin block

  END  //Germany block

UNTIL TTM = 0 OR PCS_CODE = 1
```

Fort Polk

```
IF SL=1 THEN
      BEGIN
        INC POLK_ACT1
        DEC POLK_GAIN1
      END
IF SL=2 THEN
      BEGIN
        INC POLK_ACT2
        DEC POLK_GAIN2
      END
IF SL=3 THEN
      BEGIN
        INC POLK_ACT3
        DEC POLK_GAIN3
      END
IF SL=4 THEN
      BEGIN
        INC POLK_ACT4
        DEC POLK_GAIN4
      END

POLK_ACT = (POLK_ACT1+POLK_ACT2+POLK_ACT3+POLK_ACT4)

DO

      BEGIN  //Polk block

        WAIT 1
        VALUE = RAND(1)
        INC TOS
        DEC TTM

        IF SL=1 THEN

          BEGIN  //SL1 Admin block

            IF VALUE < 0.029435 THEN
              BEGIN
                    PCS_CODE = 1
              END
            IF VALUE > 0.029435 AND VALUE < 0.076849 THEN
              BEGIN
                    INC SL
                    INC POLK_ACT2
                    DEC POLK_ACT1
              END

            IF TOS > 24 AND PCS_CODE < 1 THEN

              BEGIN  //Orders block

                    NEED1 = -1000
                    BLISS_NEED1=(BLISS_TGT1-BLISS_ACT1-BLISS_GAIN1+BLISS_LOSS1)
        GERM_NEED1=(GERM_TGT1-GERM_ACT1-GERM_GAIN1+GERM_LOSS1)
        POLK_NEED1=(POLK_TGT1-POLK_ACT1-POLK_GAIN1+POLK_LOSS1)
        HOOD_NEED1=(HOOD_TGT1-HOOD_ACT1-HOOD_GAIN1+HOOD_LOSS1)
        KOREA_NEED1=(KOREA_TGT1-KOREA_ACT1-KOREA_GAIN1+KOREA_LOSS1)
        TDA_NEED1=(TDA_TGT1-TDA_ACT1-TDA_GAIN1+TDA_LOSS1)

        IF BLISS_NEED1 > NEED1 THEN
                BEGIN
                        PCS_CODE=2
                        NEED1=BLISS_NEED1
                END
        IF GERM_NEED1 > NEED1 THEN
                BEGIN
                        PCS_CODE=3
                        NEED1=GERM_NEED1
                END
        IF POLK_NEED1 > NEED1 THEN
                BEGIN
                        PCS_CODE=-1
                        NEED1=POLK_NEED1
                END
        IF HOOD_NEED1 > NEED1 THEN
                BEGIN
                        PCS_CODE=5
                        NEED1=HOOD_NEED1
                END
        IF KOREA_NEED1 > NEED1 THEN
                BEGIN
                        PCS_CODE=6
                        NEED1=KOREA_NEED1
                END
        IF TDA_NEED1 > NEED1 THEN
                BEGIN
                        PCS_CODE=7
                        NEED1=TDA_NEED1
                END

        IF NEED1 > 5 THEN TTM =6
        IF NEED1 > 10 THEN TTM =4
        IF NEED1 > 15 THEN TTM =2
        IF NEED1 < 5 THEN
                BEGIN
                        TTM=-1
                        PCS_CODE=-1
                END

        IF PCS_CODE = 2 THEN INC BLISS_GAIN1
        IF PCS_CODE = 3 THEN INC GERM_GAIN1
        IF PCS_CODE = 5 THEN INC HOOD_GAIN1
        IF PCS_CODE = 6 THEN INC KOREA_GAIN1
        IF PCS_CODE = 7 THEN INC TDA_GAIN1

        IF PCS_CODE > 0 THEN INC POLK_LOSS1

END  //Orders block
```

```
  END //SL1 Admin block

IF SL = 2 THEN

  BEGIN  //SL2 Admin block

    IF VALUE < 0.014132 THEN
      BEGIN
            PCS_CODE = 1
      END
    IF VALUE > 0.014132 AND VALUE < 0.0272651 THEN
      BEGIN
            INC SL
            INC POLK_ACT3
            DEC POLK_ACT2
      END
    IF VALUE > 0.0272651 AND VALUE < 0.0145146 THEN
      BEGIN
            DEC SL
            INC POLK_ACT1
            DEC POLK_ACT2
      END

    IF TOS > 24 AND PCS_CODE < 1 THEN

      BEGIN  //Orders block

            NEED2 = -1000
            BLISS_NEED2=(BLISS_TGT2-BLISS_ACT2-BLISS_GAIN2+BLISS_LOSS2)
            GERM_NEED2=(GERM_TGT2-GERM_ACT2-GERM_GAIN2+GERM_LOSS2)
            POLK_NEED2=(POLK_TGT2-POLK_ACT2-POLK_GAIN2+POLK_LOSS2)
            HOOD_NEED2=(HOOD_TGT2-HOOD_ACT2-HOOD_GAIN2+HOOD_LOSS2)
            KOREA_NEED2=(KOREA_TGT2-KOREA_ACT2-KOREA_GAIN2+KOREA_LOSS2)
            TDA_NEED2=(TDA_TGT2-TDA_ACT2-TDA_GAIN2+TDA_LOSS2)

            IF BLISS_NEED2 > NEED2 THEN
                    BEGIN
                            PCS_CODE=2
                            NEED2=BLISS_NEED2
                    END
            IF GERM_NEED2 > NEED2 THEN
                    BEGIN
                            PCS_CODE=3
                            NEED2=GERM_NEED2
                    END
            IF POLK_NEED2 > NEED2 THEN
                    BEGIN
                            PCS_CODE=-1
                            NEED2=POLK_NEED2
                    END
            IF HOOD_NEED2 > NEED2 THEN
                    BEGIN
                            PCS_CODE=5
                            NEED2=HOOD_NEED2
```

```
                END
            IF KOREA_NEED2 > NEED2 THEN
                BEGIN
                    PCS_CODE=6
                    NEED2=KOREA_NEED2
                END
            IF TDA_NEED2 > NEED2 THEN
                BEGIN
                    PCS_CODE=7
                    NEED2=TDA_NEED2
                END

            IF NEED2 > 5 THEN TTM =6
            IF NEED2 > 10 THEN TTM =4
            IF NEED2 > 15 THEN TTM =2
            IF NEED2 < 5 THEN
                BEGIN
                    TTM=-1
                    PCS_CODE=-1
                END

            IF PCS_CODE = 2 THEN INC BLISS_GAIN2
            IF PCS_CODE = 3 THEN INC GERM_GAIN2
            IF PCS_CODE = 5 THEN INC HOOD_GAIN2
            IF PCS_CODE = 6 THEN INC KOREA_GAIN2
            IF PCS_CODE = 7 THEN INC TDA_GAIN2

            IF PCS_CODE > 0 THEN INC POLK_LOSS2

      END  //Orders block

    END  //SL2 Admin block

  IF SL = 3 THEN

    BEGIN  //SL3 Admin block

      IF VALUE < 0.0071226 THEN
        BEGIN
            PCS_CODE = 1
        END
      IF VALUE > 0.0071226 AND VALUE < 0.0173445 THEN
        BEGIN
            INC SL
            INC POLK_ACT4
            DEC POLK_ACT3
        END
      IF VALUE > 0.0173445 AND VALUE < 0.0198757 THEN
        BEGIN
            DEC SL
            INC POLK_ACT2
            DEC POLK_ACT3
        END

      IF TOS > 24 AND PCS_CODE < 1 THEN
```

```
BEGIN //SL3 Orders block

    NEED3 = -1000
    BLISS_NEED3=(BLISS_TGT3-BLISS_ACT3-BLISS_GAIN3+BLISS_LOSS3)
    GERM_NEED3=(GERM_TGT3-GERM_ACT3-GERM_GAIN3+GERM_LOSS3)
    POLK_NEED3=(POLK_TGT3-POLK_ACT3-POLK_GAIN3+POLK_LOSS3)
    HOOD_NEED3=(HOOD_TGT3-HOOD_ACT3-HOOD_GAIN3+HOOD_LOSS3)
    KOREA_NEED3=(KOREA_TGT3-KOREA_ACT3-KOREA_GAIN3+KOREA_LOSS3)
    TDA_NEED3=(TDA_TGT3-TDA_ACT3-TDA_GAIN3+TDA_LOSS3)

    IF BLISS_NEED3 > NEED3 THEN
        BEGIN
            PCS_CODE=2
            NEED3=BLISS_NEED3
        END
    IF GERM_NEED3 > NEED3 THEN
        BEGIN
            PCS_CODE=3
            NEED3=GERM_NEED3
        END
    IF POLK_NEED3 > NEED3 THEN
        BEGIN
            PCS_CODE=-1
            NEED3=POLK_NEED3
        END
    IF HOOD_NEED3 > NEED3 THEN
        BEGIN
            PCS_CODE=5
            NEED3=HOOD_NEED3
        END
    IF KOREA_NEED3 > NEED3 THEN
        BEGIN
            PCS_CODE=6
            NEED3=KOREA_NEED3
        END
    IF TDA_NEED3 > NEED3 THEN
        BEGIN
            PCS_CODE=7
            NEED3=TDA_NEED3
        END

    IF NEED3 > 5 THEN TTM =6
    IF NEED3 > 10 THEN TTM =4
    IF NEED3 > 15 THEN TTM =2
    IF NEED3 < 5 THEN
        BEGIN
            TTM=-1
            PCS_CODE=-1
        END

    IF PCS_CODE = 2 THEN INC BLISS_GAIN3
    IF PCS_CODE = 3 THEN INC GERM_GAIN3
    IF PCS_CODE = 5 THEN INC HOOD_GAIN3
    IF PCS_CODE = 6 THEN INC KOREA_GAIN3
```

```
            IF PCS_CODE = 7 THEN INC TDA_GAIN3

            IF PCS_CODE > 0 THEN INC POLK_LOSS3

        END  //SL3 Orders block

    END  //SL3 Admin block

IF SL = 4 THEN

    BEGIN  //SL4 Admin block

      IF VALUE < 0.0129963 THEN
        BEGIN
              PCS_CODE = 1
        END
      IF VALUE > 0.0129963 AND VALUE < 0.0208022 THEN
        BEGIN
              INC SL
              PCS_CODE = 1
        END
      IF VALUE > 0.0208022 AND VALUE < 0.0211931 THEN
        BEGIN
              DEC SL
              INC POLK_ACT3
              DEC POLK_ACT4
        END

      IF TOS > 24 AND PCS_CODE < 1 THEN

        BEGIN  //SL4 Orders block

              NEED4 = -1000
              BLISS_NEED4=(BLISS_TGT4-BLISS_ACT4-BLISS_GAIN4+BLISS_LOSS4)
              GERM_NEED4=(GERM_TGT4-GERM_ACT4-GERM_GAIN4+GERM_LOSS4)
              POLK_NEED4=(POLK_TGT4-POLK_ACT4-POLK_GAIN4+POLK_LOSS4)
              HOOD_NEED4=(HOOD_TGT4-HOOD_ACT4-HOOD_GAIN4+HOOD_LOSS4)
              KOREA_NEED4=(KOREA_TGT4-KOREA_ACT4-KOREA_GAIN4+KOREA_LOSS4)
              TDA_NEED4=(TDA_TGT4-TDA_ACT4-TDA_GAIN4+TDA_LOSS4)

              IF BLISS_NEED4 > NEED4 THEN
                      BEGIN
                              PCS_CODE=2
                              NEED4=BLISS_NEED4
                      END
              IF GERM_NEED4 > NEED4 THEN
                      BEGIN
                              PCS_CODE=3
                              NEED4=GERM_NEED4
                      END
              IF POLK_NEED4 > NEED4 THEN
                      BEGIN
                              PCS_CODE=-1
                              NEED4=POLK_NEED4
                      END
```

```
                IF HOOD_NEED4 > NEED4 THEN
                    BEGIN
                        PCS_CODE=5
                        NEED4=HOOD_NEED4
                    END
                IF KOREA_NEED4 > NEED4 THEN
                    BEGIN
                        PCS_CODE=6
                        NEED4=KOREA_NEED4
                    END
                IF TDA_NEED4 > NEED4 THEN
                    BEGIN
                        PCS_CODE=7
                        NEED4=TDA_NEED4
                    END

                IF NEED4 > 5 THEN TTM =6
                IF NEED4 > 10 THEN TTM =4
                IF NEED4 > 15 THEN TTM =2
                IF NEED4 < 5 THEN
                    BEGIN
                        TTM=-1
                        PCS_CODE=-1
                    END

                IF PCS_CODE = 2 THEN INC BLISS_GAIN4
                IF PCS_CODE = 3 THEN INC GERM_GAIN4
                IF PCS_CODE = 5 THEN INC HOOD_GAIN4
                IF PCS_CODE = 6 THEN INC KOREA_GAIN4
                IF PCS_CODE = 7 THEN INC TDA_GAIN4

                IF PCS_CODE > 0 THEN INC POLK_LOSS4

            END //SL4 Orders block

        END //SL4 Admin block

    END  //Polk block

UNTIL TTM = 0 OR PCS_CODE = 1
```

Fort Hood

```
IF SL=1 THEN
      BEGIN
        INC HOOD_ACT1
        DEC HOOD_GAIN1
      END
IF SL=2 THEN
      BEGIN
        INC HOOD_ACT2
        DEC HOOD_GAIN2
      END
IF SL=3 THEN
      BEGIN
        INC HOOD_ACT3
        DEC HOOD_GAIN3
      END
IF SL=4 THEN
      BEGIN
        INC HOOD_ACT4
        DEC HOOD_GAIN4
      END

HOOD_ACT = (HOOD_ACT1+HOOD_ACT2+HOOD_ACT3+HOOD_ACT4)

DO

      BEGIN  //Hood block

        WAIT 1
        VALUE = RAND(1)
        INC TOS
        DEC TTM

        IF SL=1 THEN

          BEGIN  //SL1 Admin block

            IF VALUE < 0.029435 THEN
              BEGIN
                    PCS_CODE = 1
              END
            IF VALUE > 0.029435 AND VALUE < 0.076849 THEN
              BEGIN
                    INC SL
                    INC HOOD_ACT2
                    DEC HOOD_ACT1
              END

            IF TOS > 24 AND PCS_CODE < 1 THEN

              BEGIN  //Orders block

                    NEED1 = -1000
                    BLISS_NEED1=(BLISS_TGT1-BLISS_ACT1-BLISS_GAIN1+BLISS_LOSS1)
        GERM_NEED1=(GERM_TGT1-GERM_ACT1-GERM_GAIN1+GERM_LOSS1)
        POLK_NEED1=(POLK_TGT1-POLK_ACT1-POLK_GAIN1+POLK_LOSS1)
        HOOD_NEED1=(HOOD_TGT1-HOOD_ACT1-HOOD_GAIN1+HOOD_LOSS1)
        KOREA_NEED1=(KOREA_TGT1-KOREA_ACT1-KOREA_GAIN1+KOREA_LOSS1)
        TDA_NEED1=(TDA_TGT1-TDA_ACT1-TDA_GAIN1+TDA_LOSS1)

        IF BLISS_NEED1 > NEED1 THEN
                BEGIN
                        PCS_CODE=2
                        NEED1=BLISS_NEED1
                END
        IF GERM_NEED1 > NEED1 THEN
                BEGIN
                        PCS_CODE=3
                        NEED1=GERM_NEED1
                END
        IF POLK_NEED1 > NEED1 THEN
                BEGIN
                        PCS_CODE=4
                        NEED1=POLK_NEED1
                END
        IF HOOD_NEED1 > NEED1 THEN
                BEGIN
                        PCS_CODE=-1
                        NEED1=HOOD_NEED1
                END
        IF KOREA_NEED1 > NEED1 THEN
                BEGIN
                        PCS_CODE=6
                        NEED1=KOREA_NEED1
                END
        IF TDA_NEED1 > NEED1 THEN
                BEGIN
                        PCS_CODE=7
                        NEED1=TDA_NEED1
                END

        IF NEED1 > 5 THEN TTM =6
        IF NEED1 > 10 THEN TTM =4
        IF NEED1 > 15 THEN TTM =2
        IF NEED1 < 5 THEN
                BEGIN
                        TTM=-1
                        PCS_CODE=-1
                END

        IF PCS_CODE = 2 THEN INC BLISS_GAIN1
        IF PCS_CODE = 3 THEN INC GERM_GAIN1
        IF PCS_CODE = 4 THEN INC POLK_GAIN1
        IF PCS_CODE = 6 THEN INC KOREA_GAIN1
        IF PCS_CODE = 7 THEN INC TDA_GAIN1

        IF PCS_CODE > 0 THEN INC HOOD_LOSS1

END  //Orders block
```

```
  END //SL1 Admin block

IF SL = 2 THEN

  BEGIN  //SL2 Admin block

    IF VALUE < 0.014132 THEN
      BEGIN
            PCS_CODE = 1
      END
    IF VALUE > 0.014132 AND VALUE < 0.0272651 THEN
      BEGIN
            INC SL
            DEC HOOD_ACT2
            INC HOOD_ACT3
      END
    IF VALUE > 0.0272651 AND VALUE < 0.0145146 THEN
      BEGIN
            DEC SL
            INC HOOD_ACT1
            DEC HOOD_ACT2
      END

    IF TOS > 24 AND PCS_CODE < 1 THEN

      BEGIN  //Orders block

            NEED2 = -1000
            BLISS_NEED2=(BLISS_TGT2-BLISS_ACT2-BLISS_GAIN2+BLISS_LOSS2)
            GERM_NEED2=(GERM_TGT2-GERM_ACT2-GERM_GAIN2+GERM_LOSS2)
            POLK_NEED2=(POLK_TGT2-POLK_ACT2-POLK_GAIN2+POLK_LOSS2)
            HOOD_NEED2=(HOOD_TGT2-HOOD_ACT2-HOOD_GAIN2+HOOD_LOSS2)
            KOREA_NEED2=(KOREA_TGT2-KOREA_ACT2-KOREA_GAIN2+KOREA_LOSS2)
            TDA_NEED2=(TDA_TGT2-TDA_ACT2-TDA_GAIN2+TDA_LOSS2)

            IF BLISS_NEED2 > NEED2 THEN
                    BEGIN
                            PCS_CODE=2
                            NEED2=BLISS_NEED2
                    END
            IF GERM_NEED2 > NEED2 THEN
                    BEGIN
                            PCS_CODE=3
                            NEED2=GERM_NEED2
                    END
            IF POLK_NEED2 > NEED2 THEN
                    BEGIN
                            PCS_CODE=4
                            NEED2=POLK_NEED2
                    END
            IF HOOD_NEED2 > NEED2 THEN
                    BEGIN
                            PCS_CODE=-1
                            NEED2=HOOD_NEED2
```

```
                END
            IF KOREA_NEED2 > NEED2 THEN
                BEGIN
                    PCS_CODE=6
                    NEED2=KOREA_NEED2
                END
            IF TDA_NEED2 > NEED2 THEN
                BEGIN
                    PCS_CODE=7
                    NEED2=TDA_NEED2
                END

            IF NEED2 > 5 THEN TTM =6
            IF NEED2 > 10 THEN TTM =4
            IF NEED2 > 15 THEN TTM =2
            IF NEED2 < 5 THEN
                BEGIN
                    TTM=-1
                    PCS_CODE=-1
                END

            IF PCS_CODE = 2 THEN INC BLISS_GAIN2
            IF PCS_CODE = 3 THEN INC GERM_GAIN2
            IF PCS_CODE = 4 THEN INC POLK_GAIN2
            IF PCS_CODE = 6 THEN INC KOREA_GAIN2
            IF PCS_CODE = 7 THEN INC TDA_GAIN2

            IF PCS_CODE > 0 THEN INC HOOD_LOSS2

      END  //Orders block

  END  //SL2 Admin block

IF SL = 3 THEN

  BEGIN  //SL3 Admin block

    IF VALUE < 0.0071226 THEN
      BEGIN
            PCS_CODE = 1
      END
    IF VALUE > 0.0071226 AND VALUE < 0.0173445 THEN
      BEGIN
            INC SL
            INC HOOD_ACT4
            DEC HOOD_ACT3
      END
    IF VALUE > 0.0173445 AND VALUE < 0.0198757 THEN
      BEGIN
            DEC SL
            DEC HOOD_ACT3
            INC HOOD_ACT2
      END

    IF TOS > 24 AND PCS_CODE < 1 THEN
```

```
BEGIN  //SL3 Orders block

     NEED3 = -1000
     BLISS_NEED3=(BLISS_TGT3-BLISS_ACT3-BLISS_GAIN3+BLISS_LOSS3)
     GERM_NEED3=(GERM_TGT3-GERM_ACT3-GERM_GAIN3+GERM_LOSS3)
     POLK_NEED3=(POLK_TGT3-POLK_ACT3-POLK_GAIN3+POLK_LOSS3)
     HOOD_NEED3=(HOOD_TGT3-HOOD_ACT3-HOOD_GAIN3+HOOD_LOSS3)
     KOREA_NEED3=(KOREA_TGT3-KOREA_ACT3-KOREA_GAIN3+KOREA_LOSS3)
     TDA_NEED3=(TDA_TGT3-TDA_ACT3-TDA_GAIN3+TDA_LOSS3)

     IF BLISS_NEED3 > NEED3 THEN
          BEGIN
               PCS_CODE=2
               NEED3=BLISS_NEED3
          END
     IF GERM_NEED3 > NEED3 THEN
          BEGIN
               PCS_CODE=3
               NEED3=GERM_NEED3
          END
     IF POLK_NEED3 > NEED3 THEN
          BEGIN
               PCS_CODE=4
               NEED3=POLK_NEED3
          END
     IF HOOD_NEED3 > NEED3 THEN
          BEGIN
               PCS_CODE=-1
               NEED3=HOOD_NEED3
          END
     IF KOREA_NEED3 > NEED3 THEN
          BEGIN
               PCS_CODE=6
               NEED3=KOREA_NEED3
          END
     IF TDA_NEED3 > NEED3 THEN
          BEGIN
               PCS_CODE=7
               NEED3=TDA_NEED3
          END

     IF NEED3 > 5 THEN TTM =6
     IF NEED3 > 10 THEN TTM =4
     IF NEED3 > 15 THEN TTM =2
     IF NEED3 < 5 THEN
          BEGIN
               TTM=-1
               PCS_CODE=-1
          END

     IF PCS_CODE = 2 THEN INC BLISS_GAIN3
     IF PCS_CODE = 3 THEN INC GERM_GAIN3
     IF PCS_CODE = 4 THEN INC POLK_GAIN3
     IF PCS_CODE = 6 THEN INC KOREA_GAIN3
```

```
            IF PCS_CODE = 7 THEN INC TDA_GAIN3

            IF PCS_CODE > 0 THEN INC HOOD_LOSS3

         END  //SL3 Orders block

      END  //SL3 Admin block

   IF SL = 4 THEN

      BEGIN  //SL4 Admin block

         IF VALUE < 0.0129963 THEN
            BEGIN
                  PCS_CODE = 1
            END
         IF VALUE > 0.0129963 AND VALUE < 0.0208022 THEN
            BEGIN
                  INC SL
                  PCS_CODE = 1
            END
         IF VALUE > 0.0208022 AND VALUE < 0.0211931 THEN
            BEGIN
                  DEC SL
                  DEC HOOD_ACT4
                  INC HOOD_ACT3
            END

         IF TOS > 24 AND PCS_CODE < 1 THEN

            BEGIN  //SL4 Orders block

                  NEED4 = -1000
                  BLISS_NEED4=(BLISS_TGT4-BLISS_ACT4-BLISS_GAIN4+BLISS_LOSS4)
                  GERM_NEED4=(GERM_TGT4-GERM_ACT4-GERM_GAIN4+GERM_LOSS4)
                  POLK_NEED4=(POLK_TGT4-POLK_ACT4-POLK_GAIN4+POLK_LOSS4)
                  HOOD_NEED4=(HOOD_TGT4-HOOD_ACT4-HOOD_GAIN4+HOOD_LOSS4)
                  KOREA_NEED4=(KOREA_TGT4-KOREA_ACT4-KOREA_GAIN4+KOREA_LOSS4)
                  TDA_NEED4=(TDA_TGT4-TDA_ACT4-TDA_GAIN4+TDA_LOSS4)

                  IF BLISS_NEED4 > NEED4 THEN
                        BEGIN
                              PCS_CODE=2
                              NEED4=BLISS_NEED4
                        END
                  IF GERM_NEED4 > NEED4 THEN
                        BEGIN
                              PCS_CODE=3
                              NEED4=GERM_NEED4
                        END
                  IF POLK_NEED4 > NEED4 THEN
                        BEGIN
                              PCS_CODE=4
                              NEED4=POLK_NEED4
                        END
```

```
                IF HOOD_NEED4 > NEED4 THEN
                    BEGIN
                        PCS_CODE=-1
                        NEED4=HOOD_NEED4
                    END
                IF KOREA_NEED4 > NEED4 THEN
                    BEGIN
                        PCS_CODE=6
                        NEED4=KOREA_NEED4
                    END
                IF TDA_NEED4 > NEED4 THEN
                    BEGIN
                        PCS_CODE=7
                        NEED4=TDA_NEED4
                    END

                IF NEED4 > 5 THEN TTM =6
                IF NEED4 > 10 THEN TTM =4
                IF NEED4 > 15 THEN TTM =2
                IF NEED4 < 5 THEN
                    BEGIN
                        TTM=-1
                        PCS_CODE=-1
                    END

                IF PCS_CODE = 2 THEN INC BLISS_GAIN4
                IF PCS_CODE = 3 THEN INC GERM_GAIN4
                IF PCS_CODE = 4 THEN INC POLK_GAIN4
                IF PCS_CODE = 6 THEN INC KOREA_GAIN4
                IF PCS_CODE = 7 THEN INC TDA_GAIN4

                IF PCS_CODE > 0 THEN INC HOOD_LOSS4

            END //SL4 Orders block

        END //SL4 Admin block

    END //Hood block

UNTIL TTM = 0 OR PCS_CODE = 1
```

Korea

```
IF SL=1 THEN
      BEGIN
        INC KOREA_ACT1
        DEC KOREA_GAIN1
      END
IF SL=2 THEN
      BEGIN
        INC KOREA_ACT2
        DEC KOREA_GAIN2
      END
IF SL=3 THEN
      BEGIN
        INC KOREA_ACT3
        DEC KOREA_GAIN3
      END
IF SL=4 THEN
      BEGIN
        INC KOREA_ACT4
        DEC KOREA_GAIN4
      END

KOREA_ACT = (KOREA_ACT1+KOREA_ACT2+KOREA_ACT3+KOREA_ACT4)

DO

      BEGIN  //Korea block

        WAIT 1
        VALUE = RAND(1)
        INC TOS
        DEC TTM

        IF SL=1 THEN

          BEGIN  //SL1 Admin block

            IF VALUE < 0.029435 THEN
              BEGIN
                    PCS_CODE = 1
              END
            IF VALUE > 0.029435 AND VALUE < 0.076849 THEN
              BEGIN
                    INC SL
                    DEC KOREA_ACT1
                    INC KOREA_ACT2
              END

            IF TOS > 8 AND PCS_CODE < 1 THEN

              BEGIN  //SL1 Orders block

                    NEED1 = -1000
                    BLISS_NEED1=(BLISS_TGT1-BLISS_ACT1-BLISS_GAIN1+BLISS_LOSS1)
        GERM_NEED1=(GERM_TGT1-GERM_ACT1-GERM_GAIN1+GERM_LOSS1)
        POLK_NEED1=(POLK_TGT1-POLK_ACT1-POLK_GAIN1+POLK_LOSS1)
        HOOD_NEED1=(HOOD_TGT1-HOOD_ACT1-HOOD_GAIN1+HOOD_LOSS1)
        KOREA_NEED1=(KOREA_TGT1-KOREA_ACT1-KOREA_GAIN1+KOREA_LOSS1)
        TDA_NEED1=(TDA_TGT1-TDA_ACT1-TDA_GAIN1+TDA_LOSS1)

        IF BLISS_NEED1 > NEED1 THEN
                BEGIN
                        PCS_CODE=2
                        NEED1=BLISS_NEED1
                END
        IF GERM_NEED1 > NEED1 THEN
                BEGIN
                        PCS_CODE=3
                        NEED1=GERM_NEED1
                END
         IF POLK_NEED1 > NEED1 THEN
                BEGIN
                        PCS_CODE=4
                        NEED1=POLK_NEED1
                END
        IF HOOD_NEED1 > NEED1 THEN
                BEGIN
                        PCS_CODE=5
                        NEED1=HOOD_NEED1
                END
        IF TDA_NEED1 > NEED1 THEN
                BEGIN
                        PCS_CODE=7
                        NEED1=TDA_NEED1
                END

        IF VALUE < 0.05 THEN
                BEGIN
                        PCS_CODE=-1
                        TTM=-1
                        TOS = 0
                        INC KOREA_LOSS1
                END

        ELSE
                BEGIN
                        IF NEED1 < 0 THEN TTM = 4
                        IF NEED1 > 5 THEN TTM = 3
                        IF NEED1 > 15 THEN TTM =2
                        IF PCS_CODE = 2 THEN INC BLISS_GAIN1

                        IF PCS_CODE = 3 THEN INC GERM_GAIN1
                        IF PCS_CODE = 4 THEN INC POLK_GAIN1
                        IF PCS_CODE = 5 THEN INC HOOD_GAIN1
                        IF PCS_CODE = 7 THEN INC TDA_GAIN1
                END

END  //SL1 Orders block
```

```
  END  //SL1 Admin block

IF SL = 2 THEN

  BEGIN  //SL2 Admin block

    IF VALUE < 0.014132 THEN
      BEGIN
            PCS_CODE = 1
      END
    IF VALUE > 0.014132 AND VALUE < 0.0272651 THEN
      BEGIN
            INC SL
            DEC KOREA_ACT2
            INC KOREA_ACT3
      END
    IF VALUE > 0.0272651 AND VALUE < 0.0145146 THEN
      BEGIN
            DEC SL
            DEC KOREA_ACT2
            INC KOREA_ACT1
      END

    IF TOS > 8 AND PCS_CODE < 1 THEN

      BEGIN  //SL2 Orders block

            NEED2 = -1000
            BLISS_NEED2=(BLISS_TGT2-BLISS_ACT2-BLISS_GAIN2+BLISS_LOSS2)
            GERM_NEED2=(GERM_TGT2-GERM_ACT2-GERM_GAIN2+GERM_LOSS2)
            POLK_NEED2=(POLK_TGT2-POLK_ACT2-POLK_GAIN2+POLK_LOSS2)
            HOOD_NEED2=(HOOD_TGT2-HOOD_ACT2-HOOD_GAIN2+HOOD_LOSS2)
            KOREA_NEED2=(KOREA_TGT2-KOREA_ACT2-KOREA_GAIN2+KOREA_LOSS2)
            TDA_NEED2=(TDA_TGT2-TDA_ACT2-TDA_GAIN2+TDA_LOSS2)

            IF BLISS_NEED2 > NEED2 THEN
                  BEGIN
                        PCS_CODE=2
                        NEED2=BLISS_NEED2
                  END
            IF GERM_NEED2 > NEED2 THEN
                  BEGIN
                        PCS_CODE=3
                        NEED2=GERM_NEED2
                  END
            IF POLK_NEED2 > NEED2 THEN
                  BEGIN
                        PCS_CODE=4
                        NEED2=POLK_NEED2
                  END
            IF HOOD_NEED2 > NEED2 THEN
                  BEGIN
                        PCS_CODE=5
                        NEED2=HOOD_NEED2
                  END
            IF TDA_NEED2 > NEED2 THEN
                    BEGIN
                            PCS_CODE=7
                            NEED2=TDA_NEED2
                    END

            IF VALUE < 0.05 THEN
                    BEGIN
                            PCS_CODE=-1
                            TTM=-1
                            TOS = 0
                            INC KOREA_LOSS2
                    END

            ELSE
                    BEGIN
                            IF NEED2 < 0 THEN TTM = 4
                            IF NEED2 > 5 THEN TTM = 3
                            IF NEED2 > 15 THEN TTM =2
                            IF PCS_CODE = 2 THEN INC BLISS_GAIN2
                            IF PCS_CODE = 3 THEN INC GERM_GAIN2
                            IF PCS_CODE = 4 THEN INC POLK_GAIN2
                            IF PCS_CODE = 5 THEN INC HOOD_GAIN2
                            IF PCS_CODE = 7 THEN INC TDA_GAIN2
                    END

      END  //SL2 Orders block

  END  //SL2 Admin block

IF SL = 3 THEN

  BEGIN  //SL3 Admin block

    IF VALUE.< 0.0071226 THEN
      BEGIN
            PCS_CODE = 1
      END
    IF VALUE > 0.0071226 AND VALUE < 0.0173445 THEN
      BEGIN
            INC SL
            INC KOREA_ACT4
            DEC KOREA_ACT3
      END
    IF VALUE > 0.0173445 AND VALUE < 0.0198757 THEN
      BEGIN
            DEC SL
            INC KOREA_ACT2
            DEC KOREA_ACT3
      END

    IF TOS > 8 AND PCS_CODE < 1 THEN

      BEGIN  //SL3 Orders block
```

```
NEED3 = -1000
BLISS_NEED3=(BLISS_TGT3-BLISS_ACT3-BLISS_GAIN3+BLISS_LOSS3)
GERM_NEED3=(GERM_TGT3-GERM_ACT3-GERM_GAIN3+GERM_LOSS3)
POLK_NEED3=(POLK_TGT3-POLK_ACT3-POLK_GAIN3+POLK_LOSS3)
HOOD_NEED3=(HOOD_TGT3-HOOD_ACT3-HOOD_GAIN3+HOOD_LOSS3)
KOREA_NEED3=(KOREA_TGT3-KOREA_ACT3-KOREA_GAIN3+KOREA_LOSS3)
TDA_NEED3=(TDA_TGT3-TDA_ACT3-TDA_GAIN3+TDA_LOSS3)

IF BLISS_NEED3 > NEED3 THEN
        BEGIN
                PCS_CODE=2
                NEED3=BLISS_NEED3
        END
IF GERM_NEED3 > NEED3 THEN
        BEGIN
                PCS_CODE=3
                NEED3=GERM_NEED3
        END
IF POLK_NEED3 > NEED3 THEN
        BEGIN
                PCS_CODE=4
                NEED3=POLK_NEED3
        END
IF HOOD_NEED3 > NEED3 THEN
        BEGIN
                PCS_CODE=5
                NEED3=HOOD_NEED3
        END
IF TDA_NEED3 > NEED3 THEN
        BEGIN
                PCS_CODE=7
                NEED3=TDA_NEED3
        END

IF VALUE < 0.05 THEN
        BEGIN
                PCS_CODE=-1
                TTM=-1
                TOS = 0
                INC KOREA_LOSS3
        END

ELSE
        BEGIN
                IF NEED3 < 0 THEN TTM = 4
                IF NEED3 > 5 THEN TTM = 3
                IF NEED3 > 15 THEN TTM =2
                IF PCS_CODE = 2 THEN INC BLISS_GAIN3
                IF PCS_CODE = 3 THEN INC GERM_GAIN3
                IF PCS_CODE = 4 THEN INC POLK_GAIN3
                IF PCS_CODE = 5 THEN INC HOOD_GAIN3
                IF PCS_CODE = 7 THEN INC TDA_GAIN3
        END
        END  //SL3 Orders block

    END  //SL3 Admin block

  IF SL = 4 THEN

    BEGIN  //SL4 Admin block

      IF VALUE < 0.0129963 THEN
        BEGIN
              PCS_CODE = 1
        END
      IF VALUE > 0.0129963 AND VALUE < 0.0208022 THEN
        BEGIN
              INC SL
              PCS_CODE = 1
        END
      IF VALUE > 0.0208022 AND VALUE < 0.0211931 THEN
        BEGIN
              DEC SL
              INC KOREA_ACT3
              DEC KOREA_ACT4
        END

      IF TOS > 8 AND PCS_CODE < 1 THEN

        BEGIN  //SL4 Orders block

              NEED4 = -1000
              BLISS_NEED4=(BLISS_TGT4-BLISS_ACT4+BLISS_GAIN4-BLISS_LOSS4)
              GERM_NEED4=(GERM_TGT4-GERM_ACT4+GERM_GAIN4-GERM_LOSS4)
              POLK_NEED4=(POLK_TGT4-POLK_ACT4+POLK_GAIN4-POLK_LOSS4)
              HOOD_NEED4=(HOOD_TGT4-HOOD_ACT4+HOOD_GAIN4-HOOD_LOSS4)
              KOREA_NEED4=(KOREA_TGT4-KOREA_ACT4+KOREA_GAIN4-KOREA_LOSS4)
              TDA_NEED4=(TDA_TGT4-TDA_ACT4+TDA_GAIN4-TDA_LOSS4)

              IF BLISS_NEED4 > NEED4 THEN
                      BEGIN
                              PCS_CODE=2
                              NEED4=BLISS_NEED4
                      END
              IF GERM_NEED4 > NEED4 THEN
                      BEGIN
                              PCS_CODE=3
                              NEED4=GERM_NEED4
                      END
              IF POLK_NEED4 > NEED4 THEN
                      BEGIN
                              PCS_CODE=4
                              NEED4=POLK_NEED4
                      END
              IF HOOD_NEED4 > NEED4 THEN
                      BEGIN
                              PCS_CODE=5
                              NEED4=HOOD_NEED4
```

```
                END
            IF TDA_NEED4 > NEED4 THEN
                BEGIN
                    PCS_CODE=7
                    NEED4=TDA_NEED4
                END

            IF VALUE < 0.05 THEN
                BEGIN
                    PCS_CODE=-1
                    TTM=-1
                    TOS = 0
                    INC KOREA_LOSS4
                END

            ELSE
                BEGIN
                    IF NEED4 < 0 THEN TTM = 4
                    IF NEED4 > 5 THEN TTM = 3
                    IF NEED4 > 15 THEN TTM =2
                    IF PCS_CODE = 2 THEN INC BLISS_GAIN4
                    IF PCS_CODE = 3 THEN INC GERM_GAIN4
                    IF PCS_CODE = 4 THEN INC POLK_GAIN4
                    IF PCS_CODE = 5 THEN INC HOOD_GAIN4
                    IF PCS_CODE = 7 THEN INC TDA_GAIN4
                END

        END  //SL4 Orders block

      END  //SL4 Admin block

    END  //Korea block

UNTIL TTM = 0 OR PCS_CODE = 1
```

TDA Units

```
IF SL=1 THEN
      BEGIN
        INC TDA_ACT1
        DEC TDA_GAIN1
      END
IF SL=2 THEN
      BEGIN
        INC TDA_ACT2
        DEC TDA_GAIN2
      END
IF SL=3 THEN
      BEGIN
        INC TDA_ACT3
        DEC TDA_GAIN3
      END
IF SL=4 THEN
      BEGIN
        INC TDA_ACT4
        DEC TDA_GAIN4
      END

TDA_ACT = (TDA_ACT1+TDA_ACT2+TDA_ACT3+TDA_ACT4)

DO

      BEGIN //TDA block

        WAIT 1
        VALUE = RAND(1)
        INC TOS
        DEC TTM

        IF SL=1 THEN

          BEGIN //SL1 Admin block

            IF VALUE < 0.029435 THEN
              BEGIN
                    PCS_CODE = 1
              END
            IF VALUE > 0.029435 AND VALUE < 0.076849 THEN
              BEGIN
                    INC SL
                    DEC TDA_ACT1
                    INC TDA_ACT2
              END

            IF TOS > 22 AND PCS_CODE < 1 THEN

              BEGIN //SL1 Orders block

                    NEED1 = -1000
                    BLISS_NEED1=(BLISS_TGT1-BLISS_ACT1-BLISS_GAIN1+BLISS_LOSS1)
        GERM_NEED1=(GERM_TGT1-GERM_ACT1-GERM_GAIN1+GERM_LOSS1)
        POLK_NEED1=(POLK_TGT1-POLK_ACT1-POLK_GAIN1+POLK_LOSS1)
        HOOD_NEED1=(HOOD_TGT1-HOOD_ACT1-HOOD_GAIN1+HOOD_LOSS1)
        KOREA_NEED1=(KOREA_TGT1-KOREA_ACT1-KOREA_GAIN1+KOREA_LOSS1)
        TDA_NEED1=(TDA_TGT1-TDA_ACT1-TDA_GAIN1+TDA_LOSS1)

        IF BLISS_NEED1 > NEED1 THEN
                BEGIN
                        PCS_CODE=2
                        NEED1=BLISS_NEED1
                END
        IF GERM_NEED1 > NEED1 THEN
                BEGIN
                        PCS_CODE=3
                        NEED1=GERM_NEED1
                END
        IF POLK_NEED1 > NEED1 THEN
                BEGIN
                        PCS_CODE=4
                        NEED1=POLK_NEED1
                END
        IF HOOD_NEED1 > NEED1 THEN
                BEGIN
                        PCS_CODE=5
                        NEED1=HOOD_NEED1
                END
        IF KOREA_NEED1 > NEED1 THEN
                BEGIN
                        PCS_CODE=6
                        NEED1=KOREA_NEED1
                END

        IF VALUE < 0.10 THEN
                BEGIN
                        PCS_CODE = -1
                        TTM=-1
                        TOS=0
                END
        ELSE
                BEGIN
                        INC TDA_LOSS1
                        IF NEED1 < 0 THEN TTM = 4
                        IF NEED1 > 5 THEN TTM = 3
                        IF NEED1 > 15 THEN TTM =2
                END

        IF PCS_CODE = 2 THEN INC BLISS_GAIN1
        IF PCS_CODE = 3 THEN INC GERM_GAIN1
        IF PCS_CODE = 4 THEN INC POLK_GAIN1
        IF PCS_CODE = 5 THEN INC HOOD_GAIN1
        IF PCS_CODE = 6 THEN INC KOREA_GAIN1

    END //SL1 Orders block

END //SL1 Admin block
```

```
IF SL = 2 THEN

  BEGIN  //SL2 Admin block

    IF VALUE < 0.014132 THEN
      BEGIN
            PCS_CODE = 1
      END
    IF VALUE > 0.014132 AND VALUE < 0.0272651 THEN
      BEGIN
            INC SL
            DEC TDA_ACT2
            INC TDA_ACT3
      END
    IF VALUE > 0.0272651 AND VALUE < 0.0145146 THEN
      BEGIN
            DEC SL
            DEC TDA_ACT2
            INC TDA_ACT1
      END

    IF TOS > 22 AND PCS_CODE < 1 THEN

      BEGIN  //SL2 Orders block

            NEED2 = -1000
            BLISS_NEED2=(BLISS_TGT2-BLISS_ACT2-BLISS_GAIN2+BLISS_LOSS2)
            GERM_NEED2=(GERM_TGT2-GERM_ACT2-GERM_GAIN2+GERM_LOSS2)
            POLK_NEED2=(POLK_TGT2-POLK_ACT2-POLK_GAIN2+POLK_LOSS2)
            HOOD_NEED2=(HOOD_TGT2-HOOD_ACT2-HOOD_GAIN2+HOOD_LOSS2)
            KOREA_NEED2=(KOREA_TGT2-KOREA_ACT2-KOREA_GAIN2+KOREA_LOSS2)
            TDA_NEED2=(TDA_TGT2-TDA_ACT2-TDA_GAIN2+TDA_LOSS2)

            IF BLISS_NEED2 > NEED2 THEN
                    BEGIN
                            PCS_CODE=2
                            NEED2=BLISS_NEED2
                    END
            IF GERM_NEED2 > NEED2 THEN
                    BEGIN
                            PCS_CODE=3
                            NEED2=GERM_NEED2
                    END
            IF POLK_NEED2 > NEED2 THEN
                    BEGIN
                            PCS_CODE=4
                            NEED2=POLK_NEED2
                    END
            IF HOOD_NEED2 > NEED2 THEN
                    BEGIN
                            PCS_CODE=5
                            NEED2=HOOD_NEED2
                    END
            IF KOREA_NEED2 > NEED2 THEN
```

```
                BEGIN
                        PCS_CODE=6
                        NEED2=KOREA_NEED2
                END

        IF VALUE < 0.10 THEN
                BEGIN
                        PCS_CODE = -1
                        TTM=-1
                        TOS=0
                END
        ELSE
                BEGIN
                        INC TDA_LOSS2
                        IF NEED1 < 0 THEN TTM = 4
                        IF NEED1 > 5 THEN TTM = 3
                        IF NEED1 > 15 THEN TTM =2
                END


        IF PCS_CODE = 2 THEN INC BLISS_GAIN2
        IF PCS_CODE = 3 THEN INC GERM_GAIN2
        IF PCS_CODE = 4 THEN INC POLK_GAIN2
        IF PCS_CODE = 5 THEN INC HOOD_GAIN2
        IF PCS_CODE = 6 THEN INC KOREA_GAIN2

    END  //SL2 Orders block

  END  //SL2 Admin block

IF SL = 3 THEN

  BEGIN  //SL3 Admin block

    IF VALUE < 0.0071226 THEN
      BEGIN
            PCS_CODE = 1
      END
    IF VALUE > 0.0071226 AND VALUE < 0.0173445 THEN
      BEGIN
            INC SL
            INC TDA_ACT4
            DEC TDA_ACT3
      END
    IF VALUE > 0.0173445 AND VALUE < 0.0198757 THEN
      BEGIN
            DEC SL
            DEC TDA_ACT3
            INC TDA_ACT2
      END

    IF TOS > 22 AND PCS_CODE < 1 THEN

      BEGIN  //SL3 Orders block
```

```
          NEED3 = -1000
          BLISS_NEED3=(BLISS_TGT3-BLISS_ACT3-BLISS_GAIN3+BLISS_LOSS3)
          GERM_NEED3=(GERM_TGT3-GERM_ACT3-GERM_GAIN3+GERM_LOSS3)
          POLK_NEED3=(POLK_TGT3-POLK_ACT3-POLK_GAIN3+POLK_LOSS3)
          HOOD_NEED3=(HOOD_TGT3-HOOD_ACT3-HOOD_GAIN3+HOOD_LOSS3)
          KOREA_NEED3=(KOREA_TGT3-KOREA_ACT3-KOREA_GAIN3+KOREA_LOSS3)
          TDA_NEED3=(TDA_TGT3-TDA_ACT3-TDA_GAIN3+TDA_LOSS3)

          IF BLISS_NEED3 > NEED3 THEN
                BEGIN
                      PCS_CODE=2
                      NEED3=BLISS_NEED3
                END
          IF GERM_NEED3 > NEED3 THEN
                BEGIN
                      PCS_CODE=3
                      NEED3=GERM_NEED3
                END
          IF POLK_NEED3 > NEED3 THEN
                BEGIN
                      PCS_CODE=4
                      NEED3=POLK_NEED3
                END
          IF HOOD_NEED3 > NEED3 THEN
                BEGIN
                      PCS_CODE=5
                      NEED3=HOOD_NEED3
                END
          IF KOREA_NEED3 > NEED3 THEN
                BEGIN
                      PCS_CODE=6
                      NEED3=KOREA_NEED3
                END

          IF VALUE < 0.10 THEN
                BEGIN
                      PCS_CODE = -1
                      TTM=-1
                      TOS=0
                END
          ELSE
                BEGIN
                      INC TDA_LOSS3
                      IF NEED1 < 0 THEN TTM = 4
                      IF NEED1 > 5 THEN TTM = 3
                      IF NEED1 > 15 THEN TTM =2
                END

          IF PCS_CODE = 2 THEN INC BLISS_GAIN3
          IF PCS_CODE = 3 THEN INC GERM_GAIN3
          IF PCS_CODE = 4 THEN INC POLK_GAIN3
          IF PCS_CODE = 5 THEN INC HOOD_GAIN3
          IF PCS_CODE = 6 THEN INC KOREA_GAIN3

      END //SL3 Orders block
```

```
  END  //SL3 Admin block

IF SL = 4 THEN

  BEGIN  //SL4 Admin block

    IF VALUE < 0.0129963 THEN
      BEGIN
            PCS_CODE = 1
      END
    IF VALUE > 0.0129963 AND VALUE < 0.0208022 THEN
      BEGIN
            INC SL
            PCS_CODE = 1
      END
    IF VALUE > 0.0208022 AND VALUE < 0.0211931 THEN
      BEGIN
            DEC SL
            DEC TDA_ACT4
            INC TDA_ACT3
      END

    IF TOS > 22 AND PCS_CODE < 1 THEN

      BEGIN  //SL4 Orders block

            NEED4 = -1000
            BLISS_NEED4=(BLISS_TGT4-BLISS_ACT4+BLISS_GAIN4-BLISS_LOSS4)
            GERM_NEED4=(GERM_TGT4-GERM_ACT4+GERM_GAIN4-GERM_LOSS4)
            POLK_NEED4=(POLK_TGT4-POLK_ACT4+POLK_GAIN4-POLK_LOSS4)
            HOOD_NEED4=(HOOD_TGT4-HOOD_ACT4+HOOD_GAIN4-HOOD_LOSS4)
            KOREA_NEED4=(KOREA_TGT4-KOREA_ACT4+KOREA_GAIN4-KOREA_LOSS4)
            TDA_NEED4=(TDA_TGT4-TDA_ACT4+TDA_GAIN4-TDA_LOSS4)

            IF BLISS_NEED4 > NEED4 THEN
                    BEGIN
                            PCS_CODE=2
                            NEED4=BLISS_NEED4
                    END
            IF GERM_NEED4 > NEED4 THEN
                    BEGIN
                            PCS_CODE=3
                            NEED4=GERM_NEED4
                    END
            IF POLK_NEED4 > NEED4 THEN
                    BEGIN
                            PCS_CODE=4
                            NEED4=POLK_NEED4
                    END
            IF HOOD_NEED4 > NEED4 THEN
                    BEGIN
                            PCS_CODE=5
                            NEED4=HOOD_NEED4
                    END
```

```
                IF KOREA_NEED4 > NEED4 THEN
                    BEGIN
                        PCS_CODE=6
                        NEED4=KOREA_NEED4
                    END

                IF VALUE < 0.10 THEN
                    BEGIN
                        PCS_CODE = -1
                        TTM=-1
                        TOS=0
                    END
                ELSE
                    BEGIN
                        INC TDA_LOSS4
                        IF NEED1 < 0 THEN TTM = 4
                        IF NEED1 > 5 THEN TTM = 3
                        IF NEED1 > 15 THEN TTM =2
                    END

                IF PCS_CODE = 2 THEN INC BLISS_GAIN4
                IF PCS_CODE = 3 THEN INC GERM_GAIN4
                IF PCS_CODE = 4 THEN INC POLK_GAIN4
                IF PCS_CODE = 5 THEN INC HOOD_GAIN4
                IF PCS_CODE = 6 THEN INC KOREA_GAIN4

            END //SL4 Orders block

        END //SL4 Admin block

    END //TDA block

UNTIL TTM = 0 OR PCS_CODE = 1
```

AIT

```
SL = 1
NEED1 = -1000
BLISS_NEED1=(BLISS_TGT1-BLISS_ACT1-BLISS_GAIN1+BLISS_LOSS1)
GERM_NEED1=(GERM_TGT1-GERM_ACT1-GERM_GAIN1+GERM_LOSS1)
POLK_NEED1=(POLK_TGT1-POLK_ACT1-POLK_GAIN1+POLK_LOSS1)
HOOD_NEED1=(HOOD_TGT1-HOOD_ACT1-HOOD_GAIN1+HOOD_LOSS1)
KOREA_NEED1=(KOREA_TGT1-KOREA_ACT1-KOREA_GAIN1+KOREA_LOSS1)
TDA_NEED1=(TDA_TGT1-TDA_ACT1-TDA_GAIN1+TDA_LOSS1)
IF BLISS_NEED1 > NEED1 THEN
      BEGIN
        PCS_CODE=2
        NEED1=BLISS_NEED1
      END
IF GERM_NEED1 > NEED1 THEN
      BEGIN
        PCS_CODE=3
        NEED1=GERM_NEED1
      END
IF POLK_NEED1 > NEED1 THEN
      BEGIN
        PCS_CODE=4
        NEED1=POLK_NEED1
      END
IF HOOD_NEED1 > NEED1 THEN
      BEGIN
        PCS_CODE=5
        NEED1=HOOD_NEED1
      END
IF KOREA_NEED1 > NEED1 THEN
      BEGIN
        PCS_CODE=6
        NEED1=KOREA_NEED1
      END
IF TDA_NEED1 > NEED1 THEN
      BEGIN
        PCS_CODE=7
        NEED1=TDA_NEED1
      END

IF PCS_CODE = 2 THEN INC BLISS_GAIN1
IF PCS_CODE = 3 THEN INC GERM_GAIN1
IF PCS_CODE = 4 THEN INC POLK_GAIN1
IF PCS_CODE = 5 THEN INC HOOD_GAIN1
IF PCS_CODE = 6 THEN INC KOREA_GAIN1
IF PCS_CODE = 7 THEN INC TDA_GAIN1
```