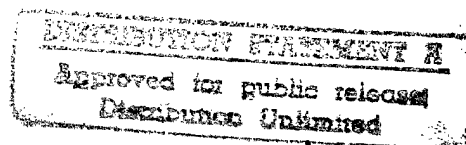# DOCUMENTED BRIEFING

RAND

New Tools for Balancing Theater Combat and Support

David Kassing, Kenneth J. Girardini,
Brian Leverich, Richard E. Stanton,
Rick Eden

Arroyo Center

19961114 026

The RAND documented briefing series is a mechanism for timely, easy-to-read reporting of research that has been briefed to the client and possibly to other audiences. Although documented briefings have been formally reviewed, they are not expected to be comprehensive or definitive. In many cases, they represent interim work.

RAND is a nonprofit institution that helps improve public policy through research and analysis. RAND's publications do not necessarily reflect the opinions or policies of its research sponsors.

DOCUMENTED BRIEFING

RAND

New Tools for Balancing Theater Combat and Support

David Kassing, Kenneth J. Girardini,
Brian Leverich, Richard E. Stanton,
Rick Eden

Arroyo Center

Approved for public release; distribution unlimited

PREFACE

This documented briefing presents an overview of tools developed to assist the Army in analyzing the effects of limitations on the size and speed of its deployments. In future deployments, force "caps" imposed by higher authorities and limitations on available air lift and sea lift could limit the rate of force closures. The resulting shortfalls, which may occur at any time during a deployment (e.g., more likely early in deployment), will tend to affect support units more than combat force closures.

The work was undertaken under the sponsorship of the Assistant Deputy Chief of Staff (Logistics) and initiated as part of the approved FY 1993 research program. The Arroyo Center was asked to examine the effects of constrained support deployments on the Army's success in accomplishing its missions. Answering this request required the integration of theater-level combat, deployment, and support modeling. The project's work has focused on integrating deployment and support modeling tools, and designing them to interface with available combat simulations. The resulting modeling process highlights the operational effects of deployment constraints.

The work should interest Army planners concerned with both combat or support operations. Army operations analysts and modelers should be interested in our observations on analytical methods and data needs.

The research was conducted in the Military Logistics Program of RAND's Arroyo Center, a federally funded research and development center sponsored by the United States Army.

CONTENTS

SUMMARY

In future contingencies, the size and speed of Army deployments to the theater could be limited. In some cases, the National Command Authority may wish to limit the size of forces put into an area. In other cases, the availability of lift or the capacities of ports in the theater could limit the rate of Army deployments.

When deployments are constrained, an important problem is to determine what support to send when doctrine cannot be satisfied. To help address this problem, we undertook to build tools to help the Army rapidly plan and replan constrained operational deployments. These tools are designed to help the Army decide what to send and when to send it; they will also help the Army articulate its needs to the Joint commanders.

THE RAND OPERATIONAL SUPPORT EVALUATOR

The project has two interrelated sets of results:

- A model for balancing constrained deployment

- Approaches to integrating the model with combat simulations

The key product of this project is a model for choosing the support to send to a theater when total deployments are constrained. We call it the ROSE model: the RAND Operational Support Evaluator. ROSE is a model that:

- allows simultaneous input of combat and support plans,

- assesses the support and deployment feasibility of a combat

  plan, and

- is potentially useful at several Army commands.

The ROSE model is formulated as a linear program. Its objective is to minimize the shortfall in support. When no shortfall is identified, the combat plan is judged supportable.

ROSE imposes four sets of constraints in each time period (e.g., a day). One set of constraints includes deployment limitations, be they a force cap, time-phased strategic lift availability, or reception capacity in theater. The second constraint simply requires that no more Army units are sent than are available. The third set of constraints describes the network capacities in the theater. The final set of constraints requires that the support requirement be satisfied. If the allocation of support units in the theater cannot provide the needed support, there is a shortfall.

ROSE offers four attractive features:

- It provides the Army the capability to evaluate the effects of deployment constraints.

- Deployment constraints and support planning are considered simultaneously.

- ROSE's results can be fed directly into combat simulations.

- Feedback loops allow for trade-offs between support and combat force deployments.

The ROSE model employs government-licensed software and, if desired, could be exportable to the Army's analytical and planning agencies and commands.

APPROACHES TO INTEGRATING ROSE WITH COMBAT SIMULATION

Differences in the structures of available combat simulations in the Army and elsewhere require different modeling interfaces. Our work has revealed three general classes of interfaces.

The first is to use ROSE output to describe the deployments for analyses with models that simulate only combat. In this case, the benefits are limited,

because a manual replanning loop is needed if the course of combat differs from the initial script or plan. Many iterations are required to approximate a solution.

The second type of interface applies to models that simulate both combat and support operations. The ROSE model results can be used to input both combat and support deployments. If combat results in the model diverge from those envisioned in the plan, then the simulation would be required to allocate the available support assets.

The most complex applications are those integrating ROSE with combat simulations that employ sophisticated combat planning algorithms. For this application, ROSE is employed twice. First, it produces a deployment schedule. Then, a second, smaller version of ROSE (one without deployment considerations) is employed to interact dynamically with the combat model's planning function to test the supportability of preferred combat plans.

Our efforts have identified several problems that modelers will face in moving further in integrating logistics and combat models.

Since combat and logistics are rarely modeled simultaneously, there are many aspects of the integration that require explicit definition. For example, the representations of combat and logistics must be compatible in several dimensions that range from the treatment of reception and onward movement to the definition of sustainment policies.

It is also vital that logical processes be compatible. The combat and support representations need consistent treatment of threshold effects as well as the ability to exchange the information needed for dynamic replanning.

The ROSE model by itself can be used to address important problems, and when it is linked to combat simulations it can become even more useful. Balancing combat and support is just one of many potential applications. The modeling process can be used to examine force planning issues, such as those arising in Force XXI, to evaluate the impacts of alternative support doctrine and programs, and to assess the effects of enemy actions against deployment operations.

1. INTRODUCTION

When deployments are constrained, an important problem is what to send when doctrine cannot be satisfied. This kind of problem was recognized following the Gulf War, when the Army had difficulty convincing the Office of the Secretary of Defense and the Joint Staff of the validity of Army support force deployment requirements.

The objective of this project is to help the Army address constrained deployment problems. We do so by providing tools for balancing combat and support deployments when constraints limit the amount of force and support that can be sent at any time. Our approach is to focus on the connections between support and combat planning and to use linear programming (LP) methods to simultaneously schedule support unit deployments and allocate support units in theater.

The tools we are developing are designed to help the Army decide what to send and when to send it. This will also help the Army articulate its needs to the Joint commanders who have the final say about deployment plans. Beyond that, the tools we are developing have the potential to address a wide range of force planning and doctrinal issues. The model will allow assessment of the effects of options that range from Force XXI restructuring initiatives to theater stockage policies. The tools we are developing are designed to be used with both existing Army combat simulations and newly developed models.

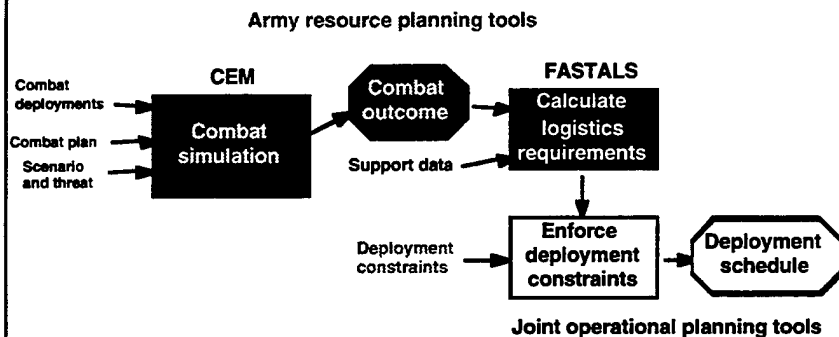**What is the Balance Between Combat and Support Deployments?**
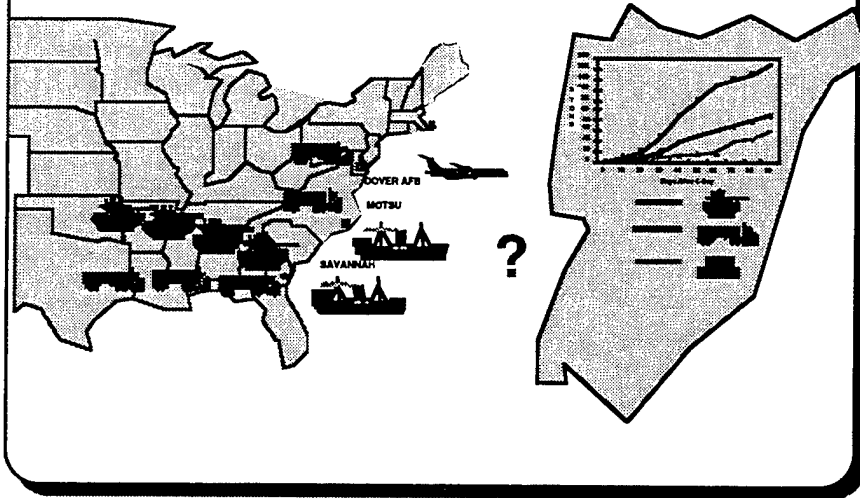
Here is a visual depiction of the problem. In peacetime, the CONUS Army is actively planning, organizing, training, and equipping forces for future contingencies. All those functions involve balancing combat and support capabilities.

But when it comes to deploying the force, the Army may not be able to send the mix of combat and support units that their planning and training have envisioned. There are three broad reasons why this may happen. First, the Army may face a "force cap" imposed by higher authority (as in Somalia). Second, strategic lift may be insufficient to sustain the desired deployment rate (as is likely to be the case at some point in any large and rapid future deployment). Third, theater facilities may limit the rate of deployment (also the case in Somalia).

The question, then, is which units to select, when to send them, and how to allocate them in the theater to get the most mission capability. The analysis must consider combat forces, support units, and supplies.

Resource Planning Tools Cannot Balance Constrained Combat/Support Deployments

Army resource planning tools

CEM — Combat simulation
- Combat deployments
- Combat plan
- Scenario and threat

Combat outcome

FASTALS — Calculate logistics requirements
- Support data

Enforce deployment constraints
- Deployment constraints

Deployment schedule

Joint operational planning tools

- **CEM and FASTALS are designed for resource planning**
- **Less useful for operational planning under constraints**

Here is a view of the analytic problem. The Army has used a set of modeling tools that work well for planning and resourcing the total Army. At the Concepts Analysis Agency, CEM and FASTALS operate as shown here in the shaded boxes. The Concepts Evaluation Model (CEM) accepts a set of combat force deployments and a plan for employing them in the theater. Other inputs include, of course, a hostile force or threat and a set of scenario data. CEM simulates combat and produces an outcome that not only tells how the battle went but also provides data relating to movement, casualties, and supply consumption. The force deployments and relevant combat results are provided to the Force Analysis Simulation of Theater Administrative and Logistics Support (FASTALS) model. FASTALS calculates the support force deployments needed to support the battle analyzed by CEM. This is all an internal Army process, aimed at identifying requirements and supporting Army program development. Note that the total deployments are not explicitly constrained through time.

Subsequently, in the Joint planning process (the white boxes in the figure), deployment constraints are imposed. CEM and FASTALS are less useful for this kind of planning. The lack of an explicit deployment constraint means they may

3

have to be run several times to approximate the constrained situation. Moreover, organizational boundaries make it difficult to communicate the results of such iterations.

Overview of Research Results

A prototype model - ROSE (RAND Operational Support Evaluator) - for balancing theater combat and support
- **Integrates combat and support planning in a single visual interface**
- **Assesses support and deployment feasibility of a combat plan**
- **Potentially useful at many Army commands**

Approaches for integrating the ROSE model with theater combat models

Implications for future theater support and combat modeling

This project yielded two types of research results for the Army. The rest of the report is organized around these two products.

The first result is an operational model for analyzing constrained deployments. We call it the ROSE model: the RAND Operational Support Evaluator. ROSE is a model that:

- allows simultaneous input of combat and support plans,

- assesses the support and deployment feasibility of a combat

    plan, and

- is potentially useful at several Army commands.

This report first presents a brief description of the ROSE model; it then provides a series of illustrative pictures in lieu of a live demonstration of the main features of the model.

5

The second result is three approaches for integrating the ROSE model with combat models. Differences in the structure of available combat simulations require that we consider different interfaces.

This work has also resulted in implications for modelers to consider in moving ahead and integrating logistics and combat models. Since combat and logistics are rarely modeled simultaneously, there are many aspects of the integration that call for compatible definitions and consistent logic.

2. THE RAND OPERATIONAL SUPPORT EVALUATOR (ROSE)

We begin by contrasting the logical flow of the ROSE model with the resource planning process recently used by the Army.

ROSE Model Changes Logical Flow and Includes Deployment Constraints
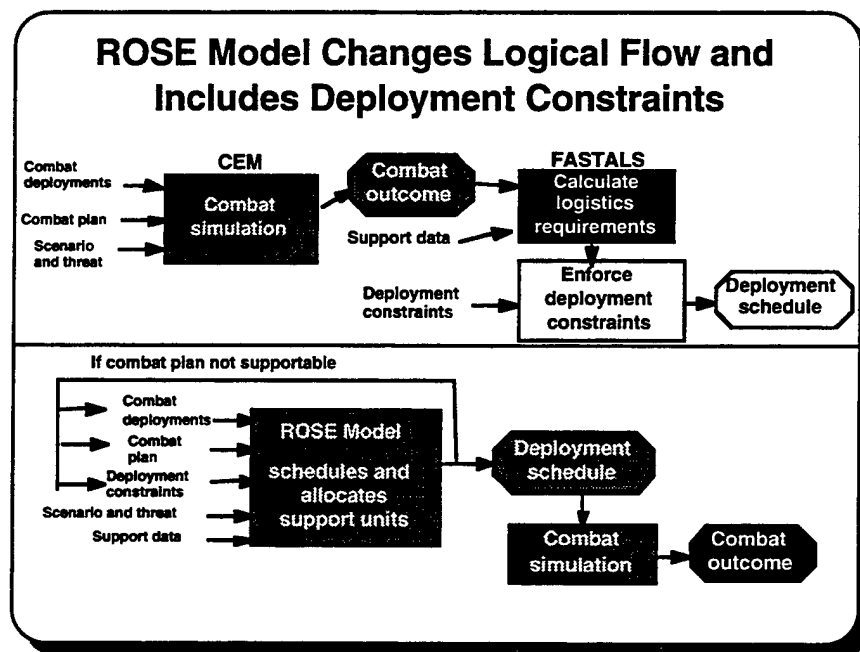
At the top of this chart, we reproduce the modeling sequence the Army uses in developing requirements. At the bottom we present the logical flow for using the ROSE model. There are four key differences:

First, ROSE is designed to allow the full operational planning process to be executed within the Army staff. We are not suggesting that ROSE should replace Joint operational planning. We are suggesting that the Army should have tools to evaluate the impacts of deployment constraints in a timely fashion, both for its own purposes and as it participates in Joint planning.

Second, the deployment constraint and logistics planning information are introduced at the start of the process. ROSE is designed to evaluate the supportability of the combat plan given the deployment constraint.

Third, ROSE produces a feasible deployment schedule that can be fed into a combat simulation. The user of the simulation can be confident that logistics resources will be available to support his combat plan. There are no restrictions on the type of combat simulation; the only requirement is that the model have some means of representing the arrival of Army units in the theater. This means that ROSE could be used with several existing combat simulations.

Fourth, there is a feedback loop if combat plans are not supportable given the deployment constraints for support units. When that occurs, one of three things (or any combination of them) must be changed. Combat force deployments can be changed (e.g., pushed back, freeing up earlier deployment resources for support) and the availability of support evaluated based on the new combat unit deployment schedule. Second, the combat plan itself can be modified to put less strain on logistics operations (e.g., a less aggressive combat plan may delay movement to contact). Third, the total deployment constraint can be lifted, i.e., the Army can get more priority for lift.

ROSE can also be used to evaluate the effects of innovations in logistics operations or policy, or to examine the impacts of new force designs, such as Force XXI.

Ultimately, ROSE yields a supportable combat plan. There is, however, no guarantee that it will be a successful combat plan, i.e., ROSE does not guarantee mission success. That is why it needs to be linked to a combat model.

The ROSE model is formulated as a linear program. The objective function in the model is to minimize the shortfall in support. When no shortfall is identified, the plan is judged supportable.

Four sets of constraints are imposed in each time period, normally a single day. One set of constraints includes the deployment limitation, be it a force cap, port capacity, or strategic lift availability. The second constraint simply requires that no more assets (e.g., units) are sent than the Army has available. The third set of constraints imposes limits based on the network capacities in the theater. The final set of constraints requires that the support requirement be satisfied. If the allocation of support units in the theater cannot provide the needed support, there is a shortfall.

Each run of the model produces a deployment schedule for support assets. The model also describes how the support assets are allocated to activities once they are deployed to the theater.

ROSE Demonstration Focuses on Army Transportation Units

Transportation
32%

Engineers
21%

Other
28%

Medical
6%

Composite
Services
13%
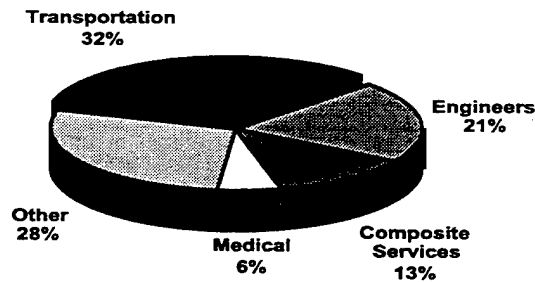
EAD deployment tonnage by branch*

*From SWA TAA 99 FASTALS results

This chart provides a simple overview of the total combat support and combat service support required by the Army for a major Southwest Asia contingency. The demonstration given in the briefing and illustrated in the next several charts focuses on transportation support in the theater. However, the modeling approach can be applied to other support units for which workload can be related to a combat plan.

We focused on transportation units because, as the chart shows, Transportation Corps units are the largest single component of Army support deployments. Heavy equipment transporters (HETs), trucks, watercraft, materiel handling equipment, and other transportation corps resources account for nearly one-third of the total. Engineer equipment accounts for just over one-fifth, and composite services (including maintenance units) provides the third-largest deployment workload.

Moreover, Transportation Corps capabilities are central to mission success in major contingencies. They unload ships, stage equipment and personnel, move units forward, and deliver sustainment to all deployed forces.

Finally, most Transportation Corps units are represented at least partially by workload-based rules in FASTALS. That means their activities can and have been modeled by the Army.

How the ROSE Model Is Used
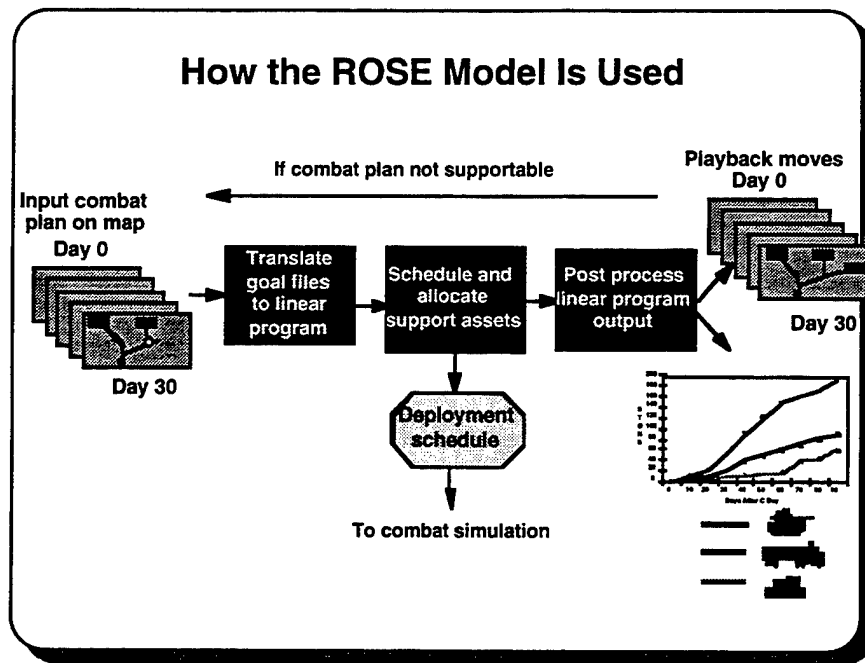
Here we depict the general process for using the ROSE model and what its outputs are.

The first step is to input the initial combat plan. In ROSE, that is accomplished using a graphical user interface (GUI). The user selects the sort of move[1] he or she wishes to make, defines the requirement, "paints" the move on a map on his screen, and records it in a data file. The process is repeated for all moves that could start on a given day (more generically, time period) and is repeated for all the days for the duration of the combat plan. The GUI speeds this process.

These GOAL (Graphic-Oriented Animation Language) files are then translated and fed into a linear program that schedules support unit deployments to theater and allocates support unit capacities within the theater.

---

[1] The term "move" is specific to transportation units. A more generic term would be task (see Appendix A).

The output of the linear program allows the user to take the resulting deployment schedule and feed its sequence into a combat model. It also allows the user to identify the sources of infeasible (or nonsupportable) plans and to play back the support operations in the theater in the same format as they were entered (i.e., graphically). Finally, results may be displayed graphically so that asset utilization and stockpiling realizations can be examined.

MapView: A Flexible GUI Adaptable to Many Models
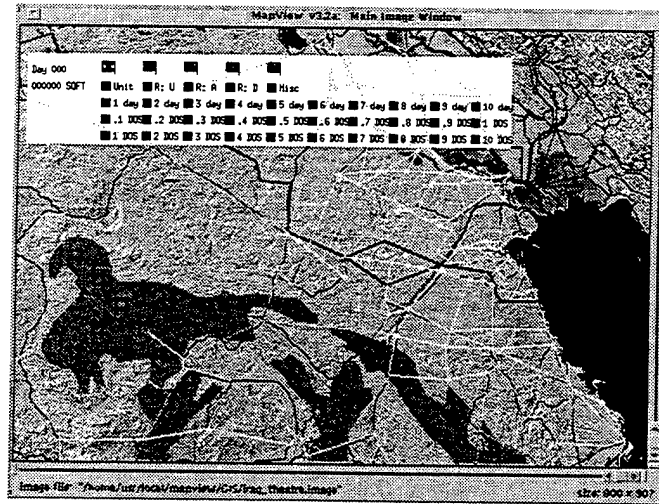
MapView provides a flexible graphical user interface that is adaptable to many combat and logistics models. It has been developed at RAND and is freely available to government agencies. For example, MapView has already been transferred to CAA.

MapView can best be thought of as a "paint" program, much like commercial drawing packages available for Macintoshes and Windows. However, it has three important capabilities that conventional paint programs don't.

First, it has a sophisticated interface to geographical information systems and can display raster, image, and vector-based maps.

Second, MapView has a rich library of military icons and other graphical objects, and the way that a user paints those objects on a map can be output in a textual GOAL file that can be easily processed for input to a model. A model's outputs can also be processed into a GOAL file that can drive animated motion of objects on the map.

Third, it is extensible. New icons, new classes of graphical objects, and even whole new areas of functionality can be added to MapView with relatively

little programming. The development of ROSE has made great use of this extensibility.

**MapView Permits ROSE Users
to Input Plans on Maps**
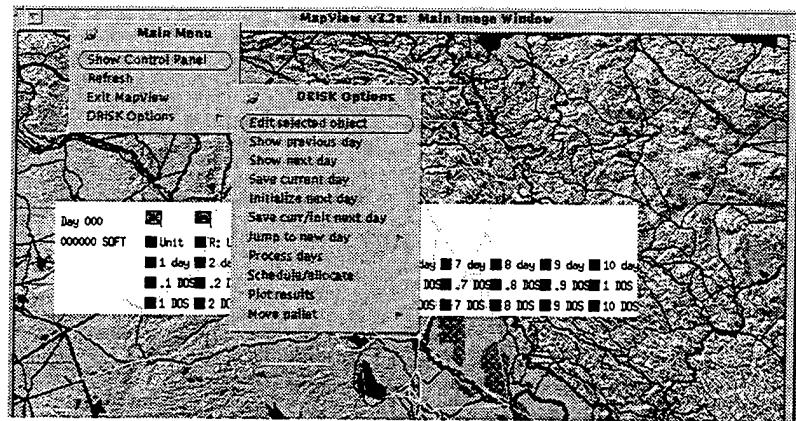
One way in which MapView is extensible is that options and new cascading menus can be added to its main menu. Those new options can either affect MapView's internal operations, or they can launch new processes on the computer upon which MapView is running

To support constrained deployment planning, we have added a submenu that allows the user to control the whole analysis process from within MapView. The first menu entry, "Edit selected object," allows users to quickly customize the characteristics of any MapView graphical object. The next six options allow a user to move back and forth through the collection of daily maps which describe the scenario. These options both affect MapView internally and also trigger separate computer processes that refine and extend the data entered by the user.

The "Process days," "Schedule/allocate," and "Plot results" options trigger the merging of the daily maps, the running of the linear programming model, and the graphical display of the results of the LP.

The "Move panel" option allows the user to reposition up, down, right, or left the "Moves" panel, which we will describe in the next chart.

The "Moves" Panel Eases the Entering of Unit and Resupply Moves

MapView v3.2a: Main Image Window

Day 000
000000 SQFT ■Unit ■R: U ■R: A ■R: D ■Misc
■1 day ■2 day ■3 day ■4 day ■5 day ■6 day ■7 day ■8 day ■9 day ■10 day
■.1 DOS ■.2 DOS ■.3 DOS ■.4 DOS ■.5 DOS ■.6 DOS ■.7 DOS ■.8 DOS ■.9 DOS ■1 DOS
■1 DOS ■2 DOS ■3 DOS ■4 DOS ■5 DOS ■6 DOS ■7 DOS ■8 DOS ■9 DOS ■10 DOS
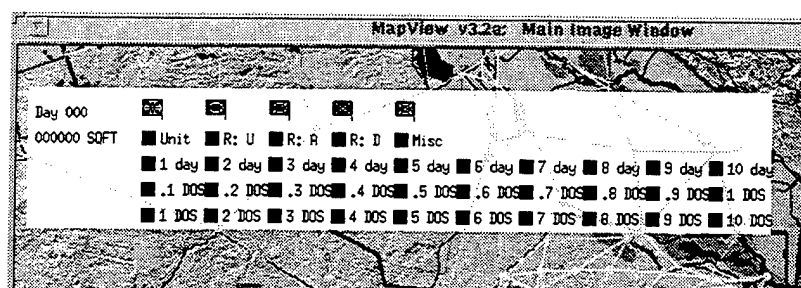
The principal reason we are using MapView is to simplify and speed the entry of unit and resupply movements.[2] The "Moves" Panel is designated to facilitate that simple and rapid data entry.

In the upper left-hand corner of the panel, you will see that we are now working on the "Day 0" map. Beneath that we show zero square feet of sealift entering the theater that day, though we could change that simply by editing that number.

The icons in the top row of the panel are representative of the types of units which a user might wish to move or resupply. Additional unit types can easily be added. The second row of buttons allows a user to specify the type of move unit, resupply (unengaged), resupply (attacking), resupply (defending), or a miscellaneous move. Other types of moves can easily be added.
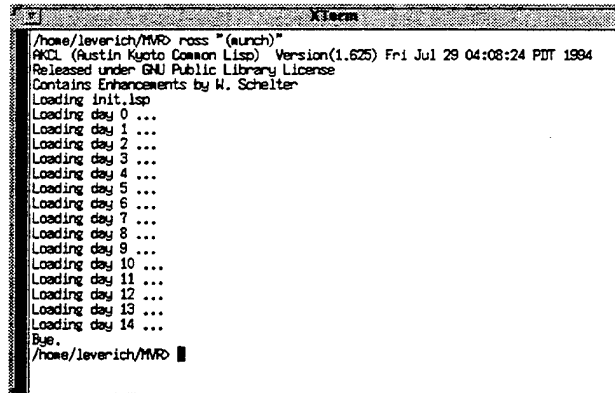
The third row specifies the "window" for the move, so long as it is between 1 and 10 days long. Longer windows can be manually entered. The fourth and fifth rows allow resupply movements to be scaled between 10% and 1000% of

---

[2] This information can also be entered or modified outside the GUI by directly editing the textual goal files.

their standard amount. The button legends read "DOS" (for days of supply), but the user is free to think of the buttons as scaling some general quantity of resupply. Moves need only be entered once; after that they are propagated across their time "window" by the software.

The "moves" panel is specialized to transportation workload. However, panels can be developed for other types of units using the extensibility of MAPVIEW (as well as the GUI, the "muncher"--see next chart--would also have to be modified to identify other types of support units). The underlying LP formulation is generic (see Appendix A), although the data used to calculate the LP coefficients and perhaps the format of the calculation would also have to be modified.

The "Muncher" Combines Daily Maps and Generates LP Matrix

```
/home/leverich/MVR> ross "(munch)"
AKCL (Austin Kyoto Common Lisp)  Version(1.625) Fri Jul 29 04:08:24 PDT 1994
Released under GNU Public Library License
Contains Enhancements by W. Schelter
Loading init.lsp
Loading day 0 ...
Loading day 1 ...
Loading day 2 ...
Loading day 3 ...
Loading day 4 ...
Loading day 5 ...
Loading day 6 ...
Loading day 7 ...
Loading day 8 ...
Loading day 9 ...
Loading day 10 ...
Loading day 11 ...
Loading day 12 ...
Loading day 13 ...
Loading day 14 ...
Bye.
/home/leverich/MVR>
```

After the sequence of daily maps containing all the unit and resupply moves has been built, the "muncher" can be called from the submenu by selecting the "Process days" menu entry.

The muncher is really several integrated pieces of software that perform two processing steps. First, an object-oriented database coded in GNU[3] Common Lisp/GNU ROSS loads and stores all the moves saved in all the daily maps, processes those moves, and finally writes an integrated summary of the moves in formats that can be used by the next processing step.

The second processing step is the execution of the matrix generator for the ROSE model's linear program. Users can configure one of several matrix generators to be automatically employed at this point in the processing. During

---

[3] "GNU" in the name of a software package usually means that the software has been released under the terms of the Free Software Foundation's General Public License. This is important to Army agencies because it means the Arroyo Center can provide the software to the Army without additional cost and the Army can freely redistribute the software. Easy transfer of the prototype ROSE model from RAND to the Army has been one of our primary design goals.

the discussion of the next slide, we will provide more detail on the different matrix generators and linear program solvers that can be employed.

**Generated LPs Are Large, But
Multiple Fast Solvers Speed Processing**
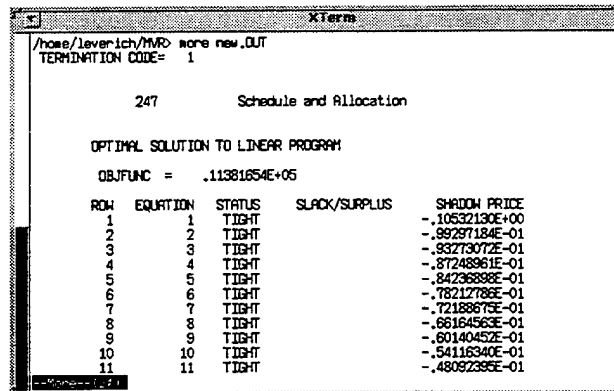
After the muncher has been run on the daily maps and the LP's matrix has been generated, the user can solve the LP by selecting the "Schedule/allocate" entry on the submenu.

The linear programs generated by the graphical inputs and the ROSE model can be large–even small deployments can generate LPs with hundreds of rows and columns, and large scenarios can easily result in LPs with thousands of rows and columns.

Because the formulation is sparse (i.e., there are relatively few nonzero coefficients) the size of the LPs is not a major concern. With modern personal computers or workstations and sparsity-exploiting LP codes, instances of the ROSE model with hundreds of rows and columns can be solved essentially instantly, and even large ROSE models can be solved in a few minutes or tens of minutes.

We have deliberately chosen to provide an open interface between the MapView GUI and the LP solver. The GUI can easily be interfaced to any solver that can read GAMS files or MPS files, or that is callable as a subroutine.

LP's Outputs Are Impenetrable, But Provide Schedule and Allocation

```
XTerm
/home/leverich/MVR> more new.OUT
  TERMINATION CODE=  1


            247          Schedule and Allocation


  OPTIMAL SOLUTION TO LINEAR PROGRAM

  OBJFUNC =   .11381654E+05

  ROW   EQUATION   STATUS   SLACK/SURPLUS      SHADOW PRICE
   1       1       TIGHT                      -.10532130E+00
   2       2       TIGHT                      -.99297184E-01
   3       3       TIGHT                      -.93273072E-01
   4       4       TIGHT                      -.87248961E-01
   5       5       TIGHT                      -.84236898E-01
   6       6       TIGHT                      -.78212786E-01
   7       7       TIGHT                      -.72188675E-01
   8       8       TIGHT                      -.66164563E-01
   9       9       TIGHT                      -.60140452E-01
  10      10       TIGHT                      -.54116340E-01
  11      11       TIGHT                      -.48092395E-01
--More--(7%)
```

The raw outputs of the linear programming model are not very user-friendly and are really only satisfactory for use by a specialist. Even for the small case illustrated here, the outputs amount to about 50 screenfuls of data.

While the raw outputs may be voluminous and difficult to understand, they do provide a complete schedule of which transportation assets arrive in the theater and when, which moves are made by which assets on which days, and what commodities cannot be moved because of shortages of transportation resources.

The nature of the raw LP outputs suggests the need for a better way for analysts to explore the schedule and allocations made by the model.

23

Graphical Outputs Ease Interpretation of the LP's Outputs
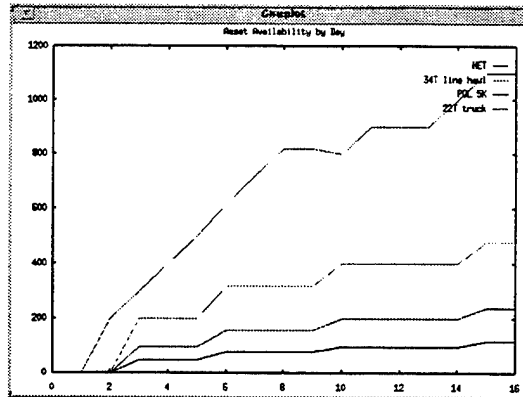
A post-processor allows the graphical presentation and rapid interpretation of various aspects of the LP's solution, including asset availability (when assets arrive in theater), asset utilization (which resources are busy when), and scheduling infeasibilities (which commodities couldn't be moved on which days).

Above you will see the graphical plot of asset availability, which clearly shows when the LP scheduled assets to arrive in the theater of operations, in this case, four types of trucks.

The graphing package is called GNU PLOT. Also, the format required for GNU PLOT can be easily imported into most spreadsheet packages.

3. INTEGRATING ROSE WITH COMBAT MODELS

The ultimate output of ROSE is both a deployment schedule determining when support units arrive in theater and an allocation of those support units to workload required to support a combat plan. To assess how well the resulting force, both combat and support, accomplishes its mission, a measure of mission success is needed. We have focused our thinking on using ROSE with combat models to illuminate the effects of logistics constraints on combat results. (In principle, ROSE could also be used for planning deployments in humanitarian operations and other operations other than war. However, since there are few models of mission performance for noncombat operations, we focused on combat missions.)

There are, of course, many kinds of combat models. This work focuses on models that deal with theater-level combat. There are at least three broad classes of theater combat simulations:

- Those that simulate combat only and disregard support

- Those that simulate both combat and support

- Those that employ dynamic combat planners

This section discusses how ROSE results can be employed with each of these types of combat models.

Simplest Approach Feeds ROSE Model Outputs into Combat Simulation

If combat plan not supportable

Combat deployments
Combat plan
Deployment constraints
Scenario and threat

ROSE Model
Schedules and allocates support units

Deployment schedule

Combat simulation

Combat outcome
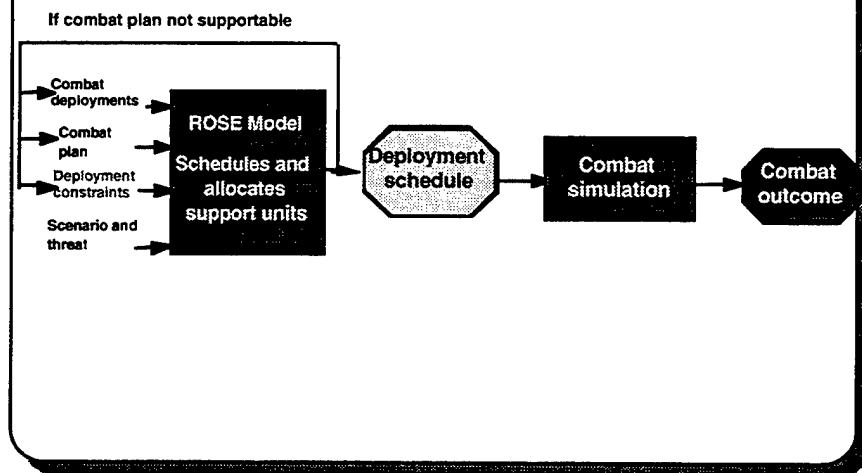
The simplest approach employs ROSE with a combat model that does not include support in any way. If the simulation examined only the single, initial combat plan, without variations resulting from operational developments, then the result could be said to be logistically supportable.

But most simulations examine what happens when combat varies from the initial plan. If the combat simulation yielded operations that differed for the combat plan studied with ROSE, the user could not be assured that the simulated combat was supportable. A cumbersome manual feedback loop would be required to insure that the actual combat that resulted in the simulation is supportable. In sum, this approach does not permit a timely dynamic assessment of the impact of support on combat outcomes.

26

A Second Approach Integrates ROSE with Combat Model That Explicitly Models Support

If combat plan not supportable

Combat deployments
Combat plan
Deployment constraints
Scenario and threat

ROSE Model
Schedules and allocates support units

Deployment schedule

Uses support deployments

Combat simulation
Support simulation

Combat outcome

Adapts to changes in combat plans
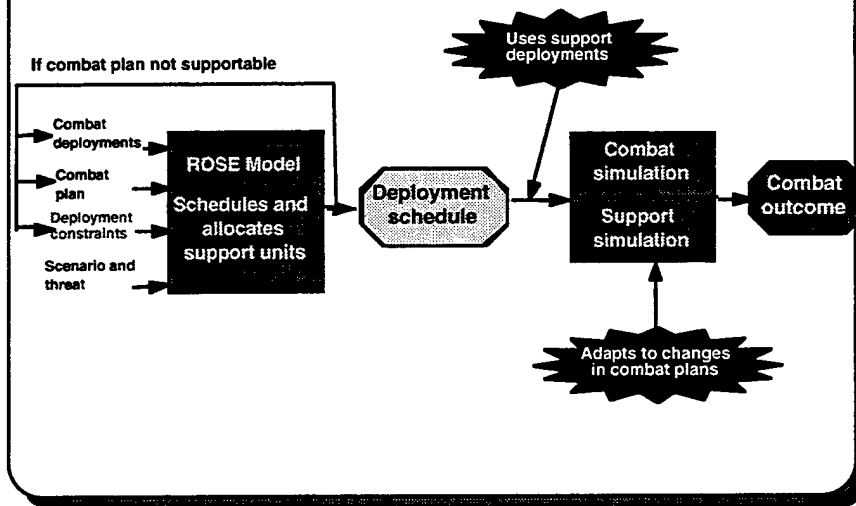
A second approach applies when both combat and support are simulated. In this modeling structure, both the combat deployment and the support deployment produced by ROSE are input to the combat simulation. If combat in the simulation varies from the initial combat plan (which it usually always does), then the explicit modeling of support forces allows the simulation to evaluate the impact of limited or misallocated support on combat outcomes. Unfortunately, these models are quite complex, and the dynamic reallocation of support is a difficult problem.

The Third Approach Integrates Modified ROSE Model with a Combat Planner
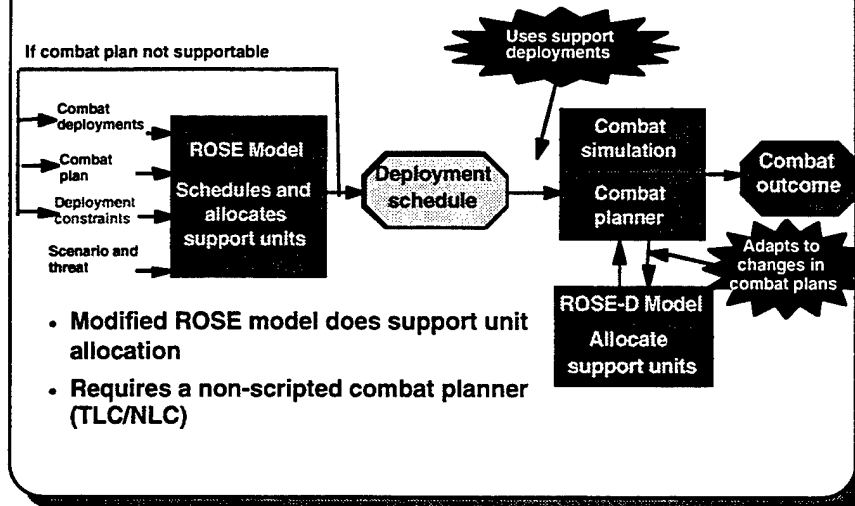
The third approach to integrating ROSE with sophisticated combat models is designed for use with combat simulations that have dynamic combat planning algorithms embedded in them. This approach has ROSE do double duty. First, it operates as we have been describing to produce a supportable plan and the associated deployment schedule. The second application of ROSE we call ROSE-D (for ROSE less the deployment routines). This is used to change support allocations dynamically within the modeling structure. Thus, when events in the combat simulation diverge from the initial plan, ROSE-D is used to adapt support allocations to maximize the capability in the new situation, given the support assets available in theater.

We have been working to use this modified ROSE model with RAND's TLC/NLC (Theater Level Campaign/Non-Linear Combat) modeling tools. The experience has surfaced some issues of more general interest for this kind of modeling integration.

28

Implications for Integrating Theater Combat and Support Models

Compatible combat and support representations
- **Movements from port to assembly areas (RSOI)**
- **Administrative and tactical unit moves**
- **Sustainment and build up of log bases**
- **Some branches difficult to represent**
- **Network flow and concept of planning**

Logical processes must also be compatible
- **Threshold effects**
- **Information to allow dynamic replanning**
- **Ability to deal with uncertainty**

Next we will briefly consider some of the general problems to be faced in linking ROSE to combat simulations.

Few models examine the dynamic interactions of combat and support. Even fewer integrate the analysis of combat and support at the theater level. In developing ROSE and considering how to link its output to combat models, we have made several observations that should benefit Army analysts and modelers as they contemplate implementing our suggestions.

Our fundamental observation is that integration requires compatible representations of support and combat operations. For example, many combat simulations initiate analysis with the combatants in place in the theater. Integration of support requires analysis of the reception and onward movement process with an explicit distinction between administrative moves (which demand support) and tactical moves (which do not). Similarly, it is necessary to model the establishment of the logistics infrastructure, including log bases, staging areas, and so on, and to assess their capacities in each time period and do this in a way that is consistent with the combat plans. For example, the representations of time and geography must be similar. And when the combat

model contains a planner, an interface must be established to link ROSE D for dynamic interaction.

Not all support can be readily represented, though all of it does make a contribution to mission success. For example, adjutant general, finance, personnel, and contracting are each difficult to assess in ROSE-like quantitative modeling. But, we note, these branches are a small part of the Army deployment demand. In other words, their needs can be satisfied without creating large problems for deployment of engineer, transportation, maintenance units, and the like.

As in all modeling and simulation work, there are threshold effects. ROSE simply adds logistics threshold problems. Quantitative thresholds may say a plan is "not supportable" when most operators would, in the event, decide that it is. For example, a combat plan may call for a log base with 15 days ammunition available at a certain location in the theater. If, because of deployment constraints and other support demands, ROSE finds that only 14.5 (or even 14.99 days) can be delivered there, then ROSE will determine that the plan is unsupportable. Assuming that the 15-day requirement is valid, there is no modeling solution to this problem. Examination of the unsupportable plans can reveal cases where deployed Army capability is very close to requirements (like 14.99 days of supply), but this is tantamount to changing the requirement and does not really do away with the threshold problem.

Finally, there is the problem of the criterion for selecting a single deployment schedule ("realization") from ROSE analysis when there are great uncertainties about the actual course of combat. Each ROSE run yields a preferred deployment schedule for the exact assumptions made and conditions modeled. If many runs are made (for example, varying only enemy objective and capabilities), many different deployment schedules may result. The problem for Army planning is to find a "robust deployment schedule," one that works quite well across most potential sets of enemy objectives and capabilities.

4. POTENTIAL POLICY ISSUES THAT CAN BE ADDRESSED BY ROSE

ROSE Model Has the Potential to Address Many Policy Issues

How should support and combat deployments be balanced?

What are the support and deployment implications of Force XXI?

What are the effects of different support policies?

- **Support from afar**
- **Improvements in distribution response times**
- **Theater stockage policy**
- **LOGCAP**

What are the effects of disrupted deployments?

But VV&A issues still must be addressed.

We think that ROSE by itself can be used to address important problems and when linked to combat simulations can become even more useful. This presentation has stressed how ROSE balances combat and support. But that is just one of many potential applications.

By changing combat force deployments, ROSE can be used to examine how combat force changes (such as those contemplated in Force XXI) would affect support and deployment needs. By changing the support force inputs (cargo requirements, unit productivities, etc.) the effects of changes in Army support doctrine and programs can be evaluated. Finally, the model can be used to show the impact of enemy actions to disrupt deployment operations and the payoff from improved defenses.

Before ROSE is used for policy analysis, there are verification, validation, and accreditation (VV&A) issues. VV&A are essential steps in model

development. Since the application of ROSE in an integrated analysis process has three parts, we address ROSE VV&A in three parts as well.

The linear programming routines that perform ROSE's analytical work are commercially available and widely used. Within RAND they have previously been applied in Arroyo projects. We argue that this record is sufficient to indicate that the linear programming routines have met verification and validation tests.

ROSE is intended to be used with several types of combat models. Many of those models have been separately verified and validated by the Army. Any new combat models (such as EAGLE or TLC/NLC) certainly require similar review and accreditation. But that is a separate process.

VV&A of ROSE should focus on the representation of Army logistics processes in ROSE. Are the representations of transportation networks and their capacities valid? Does the modeling of logistics demands (combat consumption, support base stocks, equipment failure rates, etc.) accord with recent Army experience and the results of validated higher-resolution models now in use by the Army? Is the resulting allocation of support resources realistic?

APPENDIX A: MATHEMATICAL FORMULATION

The purpose of this appendix is to describe the linear programming (LP) formulation that lies at the heart of the ROSE model. Much of the data needed to run the model is entered through the map-based graphical user interface (GUI). Other logistics planning data is entered through spreadsheets. Software translates the data from these various sources into a format acceptable to a variety of LP solvers. The translation is governed by the LP formulation presented below. Our discussion begins at a high level of aggregation and then gets increasingly specific.

General Description

Figure A.1 below provides a general description of the linear programming formulation. The objective function in the model is to minimize the shortfall in support. Shortfalls are calculated by comparing the support requirements in theater (driven by the combat plan and logistics planning factors) with the capability to deploy and allocate support assets in theater (which are the decision variables in the LP). Support shortfalls are summed over all support assets and activities. When no shortfall is identified, the combat plan is judged supportable. The output of the model is a deployment schedule specifying when support assets are deployed into theater and how they are allocated once in theater (from arrival until the end of the planning horizon).

Five categories of constraints are imposed. Each category can represent a large number of similar constraints as they are applied across multiple indices (e.g., most of the constraints are applied for each time period in the formulation).

[1][1] Deployment constraints are applied for each time period (e.g., day). Hence, if there are 30 time periods and constraints are applied for sea lift, air lift, and personnel in theater (force cap), then the deployment constraint category has 90 constraints.

---

[1] The numbers in brackets refer to the equation numbers in the formulation below.

```
Objective
    • Minimize support shortfall

Constraints (imposed each time period)
    • Deployment ≤ strategic lift / port capacity / force cap
    • Support assets deployed ≤ total available support assets
    • Intratheater network traffic ≤ arc capacity
    • Continuity constraint allocated assets ≤ deployed assets
    • Required - allocated support = support shortfall

Outputs
    • Deployment schedule for support assets
    • Allocation of support assets theater
```

Figure A.1--Summary of linear programming formulation

[2] Support asset availability is also applied for each time period (e.g., to represent the activation of reserve units) and for each type[2] of support asset. The purpose of the constraint is to insure the model does not deploy more assets than exist or are available for deployment. The formulation is generic, so support assets can represent anything from support units (e.g., medium truck companies) to specific end items (e.g., M39A trucks). The user determines the definition of assets via the input data. Hence, a problem dealing with the deployment to and allocation in theater of seven key types of trucks over 30 time periods would have 210 constraints in the support asset availability category.

[3] Assets are scheduled for deployment and allocated in theater to support a combat plan. The intratheater network traffic constraint is used to ensure the allocation of deployed assets in theater is feasible. This constraint is used whenever assets working in theater on different tasks share common nonconsumable resources in the theater. For example, the allocation of transportation assets across the theater road network to achieve a set of movements (i.e., to avoid deploying more transportation assets to theater than the road network in theater can handle). This

---

[2] Like assets can be split to represent differences in lift requirements. For example, similar assets that are prepositioned, from Europe, or from CONUS can be treated separately to differentiate lift requirements or regional asset availability.

constraint is enforced for each arc of the intratheater network and time period. Hence, for a theater with 50 key road arcs and 30 time periods there would be 1500 constraints. Consumable resources in theater may also be implemented with a similar constraint (not included in the formulation below). For consumables, capacity over time is an accumulation of available resources (e.g., inventories, deliveries, and production in theater) minus commitment (e.g., allocating resources to a scheduled task) of resources in previous time periods. An example would be gravel which can affect engineering tasks for road building, potable water supplies which can affect tasks for water transport, petroleum supplies which can affect tasks for the transport of petroleum products, etc.

[4] Because the model both decides on support assets to deploy to theater and allocates assets once they are in theater, a category of constraints is required to insure only those assets that have been deployed to theater are allocated to support activities in theater. We refer to this as a continuity constraint. It is applied for each type of asset and each time period. This constraint is also used to balance assets for which there is no deployment decision (e.g., host nation support, forward positioned, or prepositioned assets) .

[5] The final constraint category tracks the shortfall between the support required by the combat plan and the allocated support. The slack variable that balances this constraint is minimized in the objective function. The shortfall is calculated for all types of support assets and tasks.

Mathematical Formulation

The major components of the formulation are assets (indexed by i) which carry out activities (indexed by j) that are bundled together as tasks (indexed by k). Tasks have an earliest begin time and a latest completion time. That is, all activities associated with a task must be completed within the assigned window of opportunity, or a support shortfall exists. While one could define each activity as its own task (i.e., collapsing indices j and k into a single index), the current formulation simplifies the use of the GUI and aligns well with typical support requirements in theater.

Assets, activities, and tasks can be represented at any desired level of aggregation with the limitation that the LP formulation is continuous so noninteger solutions are likely. For example, if medium truck companies are represented as an asset, then the LP may assign fractions of medium truck companies to tasks or deploy fractions of medium truck companies over time. We have not investigated the

35

implications of changing the formulation from a linear program to an integer program, but the computational complexity would likely make such a change infeasible.

The interpretation of assets, activities, and tasks is best aided by an example. For transportation, tasks may be the moving of a combat unit, transporting supplies to establish inventories (e.g., X days of supply) at a log base, or moving supplies forward from a log base to units. Each of these tasks must be completed within some window of opportunity. Activities within tasks are driven by the need to differentiate the capabilities of assets in completing portions of a task. For example, the kth support task may be the movement of a brigade to a forward area of operation. Such a task may require the movement of different commodities like unit equipment, personnel, supplies, and bulk fuel for which different assets have different capabilities (some of which may be exclusive). Hence, the kth task may be broken out into numerous activities like moving tanks, fuel, containers, etc. If the ith asset is HET companies or HETs (depending on the aggregation chosen by the user), then the ith asset is the only asset that would have a positive productivity in the jth activity involving transporting tanks in a unit move (HETs can also move other types of equipment and containers).

Other indices include t for time period and r for arcs to the intratheater network (e.g., road segments for transportation tasks). The set of indices with transportation examples in parenthesis are listed below:

<u>Indices:</u>

i = assets (e.g., HETs or HET companies or medium trucks or medium truck companies) $1,....,I$

j = activities (e.g., move a tank, move bulk fuel, or move a container) $1,....,J$

k = tasks (e.g., move a brigade to a forward position, establish stocks at a log base) $1,....,K$

r = arcs for the intratheater network (e.g., road segments) $1,....,R$

t = time periods (e.g., days) $1,....,T$

As well as indices there are two sets used in the formulation. Sets represent a subset of indices for which a given condition applies.

<u>Sets:</u>

K_t = set of tasks that could be in progress during day t (subset of the set $\{k=1 \ldots K\}$).

K_r = set of tasks that involved the rth arc of the intratheater network (subset of the set {k=1 ... K}).

The formulation includes two decision variables. The first represents when assets are deployed to theater. The second determines how deployed assets are allocated to tasks and activities in theater over time. Because of their interrelated definitions, the decision variables are necessarily related by a continuity constraint. There is also a calculated variable representing the support shortfall by task and activity. The support shortfall compares support requirements (determined by the combat plan and the logistics planning factors) with support allocated (determined by the deployment and intratheater network constraints).

Variables: (all nonnegative)

$ASSET_{it}$ = number of type i assets arriving to theater in time period t. This variable determines what gets deployed into theater (e.g., like a TPFDD). Assets represent resources deployed to theater for carrying out activities (e.g., HET, HET company, bulldozer, horizontal construction company). Assets do not represent resources like fuel and spare parts that are consumed in theater.

$ASSIGN_{ijkt}$ = number of type i assets assigned at time period t to activity j of task k. This variable involves the assignment of assets deployed to theater to specific task in theater (e.g., as would be done by the logistics commander). This is not a cumulative variable and is positive only in the period that the assignment is made. Inherent in the formulation, therefore, is the assumption that once an asset is assigned to a task it will remain on that task until the task is completed. (This assumption is necessary to calculate the productivity parameter and is one of the major assumptions resulting from the LP formulation).

S_{jk} = Shortfall of activity j on task k.

Coefficients:

$RQMT_{jk}$ = number of units of activity j required to complete task k (the units of measure are a function of the type of activity). For example, this may be the number of tanks or other tracked vehicles, gallons of fuel, or containers required to move a unit 50 km over a particular type of road surface (i.e., requirement may be in tank km's).

$PROD_{ijkt}$ = productivity of type i asset assigned at time t to carry out activity j on task k. Represents the ability of an asset to contribute to a specific support

requirement. For example, the productivity of a HET may be one tank move a day (assuming the time period is a day) for the activity of transporting M1A1s associated with the task of moving an armored brigade over 300 km of paved roads. This parameter may be calculated from more detailed parameters stored in the data structure (e.g., velocity on road types, hours of driving per day, maintenance factors, etc.).

C_{kj} = weighting factor which measures the "relative importance" of activity j on task k. The default value is constant for all k and j, but it is large enough to ensure that minimizing the support shortfall takes priority (see the objective function below). However, the user may assign other values to prioritize possible support shortfalls (e.g., giving combat unit moves greater emphasis over support unit moves and building logistics bases).

$DEPUSE_i$ = deployment resource usage by asset of type i. For example, for a sealift constraint, this may represent the square feet (i.e., footprint) of an asset including a stowage factor. For an airlift constraint, this may be tons. And for a force cap constraint, it may be the number of personnel associated with asset of type i.

$DEPCAP_t$ = deployment capacity available to deliver assets to theater on day t. For example, this may be the square feet of sealift arriving in theater on a daily basis.

MAX_ASSET_{ti} = number of type i assets available at time t.

EST_k = earliest start time for task k.

LCT_k = latest completion time for task k.

CAP_{tr} = capacity of the rth arc of the intratheater network (t really denotes the minimum of t and LCT_k). Note that intratheater arc capacities can change over time (e.g., engineers can build or improve roads).

HNS_{it} = number of host nation support assets of type i "arriving" in theater during time period t.

TWF_t = time weighting factors used to place emphasis on the use of early deployment assets.

Constraints:

[1] Deployment < sea lift/air lift/force cap (shown below for sealift only)

$$\sum_{i=1}^{I} DEPUSE_i \times ASSET_{it} \le DEPCAP_t \qquad\qquad \forall t$$

[2] Support assets deployed < total support assets available

$$\sum_{q=1}^{t} ASSET_{iq} \le MAX\_ASSET_{ti} \qquad\qquad \forall i,t$$

[3] Road network traffic < arc capacity

$$\sum_{i=1}^{I} \sum_{j=1}^{J} \sum_{k \in K_t \cap K_r} ASSIGN_{ijkt} \le CAP_{tr} \qquad\qquad \forall r,t$$

[4] Continuity constraint: allocated assets = deployed assets

$$\sum_{j=1}^{J} \sum_{k \in K_t} \sum_{q=EST_k}^{t} ASSIGN_{ijkq} \le \sum_{q=1}^{t}(ASSET_{iq} + HNS_{iq}) \qquad\qquad \forall i,t$$

[5] Required - allocated support = support shortfall

$$RQMT_{jk} - \sum_{i=1}^{I} \sum_{t=EST_k}^{LCT_k} PROD_{ijkt} \times ASSIGN_{ijkt} = S_{jk} \qquad\qquad \forall j,k$$

Objective Function:

$$\text{Minimize} \quad \sum_{j=1}^{J} \sum_{k=1}^{K} C_{kj} \times S_{jk} + \sum_{i=1}^{I} \sum_{t=1}^{T} TWF_t \times DEPUSE_i \times ASSET_{it}$$

The objective function has two components: (1) weighted support shortfall and (2) deployment resources. The second term in the objective function forces the LP to choose a deployment schedule that uses the least lift when there are multiple ways to meet all the support requirements (the coefficient TWF_t should be assigned values that decrease as it increases to place a premium on early deployment resources). This is particularly important since not all support requirements will lend themselves to requirements tied directly to a combat plan (e.g., chaplains). Hence, it may ultimately be necessary to allocate lift to support assets that are not modeled in the LP. The weighting factor on the support shortfall must be large enough to assure that minimizing the support shortfall is the primary consideration and the deployment term is used only as a tie-breaker.

The LP formulation results in two major assumptions. The first is that assets once assigned to an activity are not relinquished until the activity is completed (see the ASSIGN decision variable above). A second related assumption is that once an

activity is completed, all the assigned assets are available the next time period (a constant delay can be added).[3]

There are two ways to extend the formulation to account for interdependencies between support workloads. If problem size (function of the computing platform) is a problem, the overall LP can be solved in stages. For example, assignment of engineering assets to build a pipeline (leveraging the extensibility of MAPVIEW, these tasks could be entered in through the map-based GUI) would result in a transportation requirement. This would require engineering assets to build the pipeline, but also would require a task so that transportation assets are assigned to move the engineering assets and material that cannot be moved organically prior to the beginning of work. This can be accommodated by solving the LP first for engineering requirements. This solution then generates requirements for a second run of the LP dealing with transportation assets. When solving the problem in stages, a premium on the use of early deployment capacity using weighting factors must be used in the objective function to ensure that deployment assets are allocated efficiently across all types of support assets (stages of the problem). Solving the problem in stages will also result in any support shortfalls occurring with assets and tasks in the last stage.

Interdependencies between support workloads can also be accommodated in a single LP (versus the multi-stage approach described above). This could be done by adding a term to constraint [5] that increases the support requirement for some assets based on the *ASSIGN* variable associated with other support assets. A standard move window would have to be assumed. The larger the window, the less flexibility there is with the engineering assets (they would then be unavailable). The shorter the window, the less flexible the workload for transportation assets.

---

[3] This is not to imply that support assets are always available. The productivity parameter should include maintenance factors and other considerations affecting downtime (see example).

APPENDIX B: A TRANSPORTATION EXAMPLE

We now provide a simple example of the LP model described in Appendix A. In the example a combat plan involving four unit moves must be supported. To keep the example simple we deal only with the unit moves themselves (i.e., no moves associated with resupply materials are included). Also, the only support assets involved are transportation assets.

We first describe how the problem would be entered on the GUI. Then we provide the data for the coefficients discussed in the formulation. The *PROD* coefficient for transportation assets is developed from more detailed parameters.

The results are given for a base case which includes support shortfalls. Then three options are investigated for dealing with the support shortfall.

Using the Map-Based GUI

The combat plan involves moving four divisions into defensive positions from a port area. The analyst would use the map-based GUI to enter the unit moves on the map beginning on the earliest start time (EST). The maximum time entered by the user for the move (task) then defines the latest completion time (LCT). After the user has defined the move type using menus on the map, the arcs of the road network associated with the move are highlighted.

After the user has entered all four unit moves on the appropriate maps for each EST, four tasks will have been defined ($K=4$). When the map based input is passed to the "muncher" program, the "muncher" program accesses data identifying the different kinds of commodities that must be transported for each unit move (input through spreadsheets or ASCII files). In this example we will identify four activities ($J=4$) associated with each unit move (task). The four activities are associated with transporting four different commodities: heavy equipment, 40-foot containers, POL, and 20-foot containers. To keep the example simple, no POL commodity is used in the unit moves. Four different types of support assets ($I=4$) are defined: HET, 34-ton truck, 5,000-gallon POL tanker truck, and a 22-ton truck.[1]

The "muncher" program also identifies the arcs highlighted by the user for each move and calculates the associated distances. The distances, both on-road and off-

---

[1] As stated in Appendix A, the commodities are defined to differentiate the capabilities of the different support assets.

41

road (characteristics associated with the arcs) are used in the calculation of the PROD coefficient.

In the example below, the first unit move is a hypothesized light division followed by three armored division moves. All moves were initiated at the port based on arrival times in theater.

Sample Values for the Parameters

The tables below provide specific examples of data for the LP parameters. Some of the values are passed in directly from spreadsheets while others are derived from the map-based GUI.

Requirements -- RQMT$_{jk}$. The parameter $RQMT_{jk}$ represents the number of units of activity j required to complete task k. Data is provided for each type of task represented in the on-screen menus using a spreadsheet. Users can also define unique tasks on-line using the menu input. In this example, the data below was tied to the icons selected using the GUI.

	Activity (Commodity Loads)			
Task (move)	Heavy Equipment	40-foot Containers	POL	20-foot Containers
MV1	122	81	0	65
MV2	1222	814	0	651
MV3	1222	814	0	651
MV4	1222	814	0	661

Earliest start and latest completion times -- EST$_k$ and LCT$_k$. The earliest start time is defined by the date the user enters the task on the map-based GUI. A maximum time allotted for the task then defines the LCT.

Task (move)	Earliest start time (day)	Latest completion time (day)
MV1	2	10
MV2	16	22
MV3	20	26
MV4	35	41

Asset Productivity -- PROD$_{ijkt}$ and Associated Parameters. The parameter $PROD_{ijkt}$ represents the productivity of an asset of type i assigned at time t to carry out activity j on task k. The methodology below is specific to transportation assets and is coded into the "muncher" program. For transportation assets, we estimated the

productivity of assets by first looking at the time interval in which an activity should be completed (i.e., between its earliest start time and latest completion time), determining the number of deliveries of a particular commodity (the activity -- e.g., HET loads, containers, gallons POL) that a transportation asset could make in that time interval (partial deliveries do not count), and then calculating the productivity as the product of that asset's transportation movement capability in a single movement and the number of deliveries that asset can produce in the time interval for the task at hand. The productivity of assets are calculated for each day between the EST and the LCT. The following algorithmic description summarizes the calculation:

For each task k, and for each asset i, and for each activity (commodity) j:

Let est = earliest start time for task k

Let lct = latest completion time for task k

For all t between est and lct,
   (1) Determine if asset i is appropriate for this activity j: there is a movement requirement, the ith asset has positive productivity for the jth activity, and the asset has a greater than zero movement rate, either on-road and/or off-road).

   (2) Determine movement times:
            (a) The on-road time is the on-road movement distance required divided by the asset on-road movement rate.
            (b) The off-road time is the off-road movement distance required divided by asset off-road movement rate.
            (c) The one-way time is the sum of the on-road time, off-road time, commodity load time and commodity unload time.
            (d) The round trip time is the sum of the one-way time and an unloaded return trip (on-road time and off-road time).

   (3) Determine if the time remaining between t and lct permits the completion of at least a single movement. If there is insufficient time, then $PROD_{ijkt}$ is treated as zero. If there is sufficient time, compute:
            (a) remaining time available for the movement = $lct - t$

43

(b) number of deliveries = integer part of: ((remaining time - one-way trip time)/(round trip time)) + 1) * fraction of time that the transportation asset can be utilized.

(4) Determine the productivity: $PROD_{ykt}$ is the product of the number of deliveries during the time remaining and asset i's load-carrying capability in activity (commodity) j.

An example of the *PROD* parameter is shown below for a HET moving a heavy equipment (e.g., M1 tank), on the second unit move. The table reflects that the productivity of a HET assigned to the second unit move on day 16 is 8 heavy equipment moves (which reflects the round trip distances, speed the HET operates at, load and unload times, etc.). If the same HET is assigned on day 18 it can only complete 5 moves. Since partial moves do not count (a round trip must be completed), the productivity parameter is not necessarily linear with time. Also, we have assumed that the productivity on the day of the LCT is zero (see EST and LCT above). That is, the task must be completed prior to the LCT.

	C16	C17	C18	C19	C20	C21	C22
$PROD_{112t}$	8	7	5	4	3	1	0

The following tables show examples of the various input data required to compute productivity of assets. Heavy equipment transporters (HETs) are capable of moving one heavy equipment load, or one 40-ft container, or two 20-ft containers in a single lift. The first movement is to cover 76 on-road km. Each type of movement asset is assumed to be capable of moving a total distance of 720 km per day on-road. Loading times are assumed to take one hour; unloading times 1/2 hour. Finally, we assume that none of our movement assets can be used for more than 18 hours per day.

Asset Capability – CAPABILITY$_k$. Asset capability represents the capability of each asset with respect to the activities. Note that HETs are the only asset capable of lifting heavy equipment, but that they can also be used to move both 40-ft and 20-ft containers.

44

	Activity (Commodity Loads)			
Asset	Heavy Equipment	40 foot Containers	POL	20 foot Containers
HET	1	1	0	2
34-Ton Truck	0	1	0	2
5K-Gal POL Tanker	0	0	1	0
22-Ton Truck	0	0	0	1

Movement Distance -- MOVEKM$_k$. This coefficient is calculated by the muncher based on the arcs highlighted by the user when inputting on the map based GUI and a data file listing distances and labels associated with the arcs. In the example, each unit is moved from the port varying distances to defensive positions or to be held in reserve.

	Movement Length (Km)	
Task (move)	On-road	Off-road
MV1	76	0
MV2	168	0
MV3	21	0
MV4	227	0

Asset Velocity -- VELOCITY$_i$.

	Velocity (Km per Day)	
Asset	On-road	Off-road
HET	720	0
34-Ton Truck	720	360
5K-Gal POL Tanker	720	360
22-Ton Truck	720	360

Load and Unload Times -- LUTIME$_i$

	Time in Days	
Asset	Load	Unload
HET	0.042	0.021
34-Ton Truck	0.042	0.021
5K-Gal POL Tanker	0.042	0.021
22-Ton Truck	0.042	0.021

Usage factors -- USE$_i$

Asset	Fraction of Day Asset Available
HET	0.75
34-Ton Truck	0.75
5K-Gal POL Tanker	0.75
22-Ton Truck	0.75

Deployment Resource Usage -- DEPUSE$_i$. This coefficient represents the amount of deployment resources used by an asset of type *I*. We also include here a stowage factor for each asset. Total deployment capacity used by the asset is the deployment square footage divided by the stowage factor (we used 0.75 as the stowage factor). We used the following data in our example, which does not reflect the stacking of trailers:

Asset	Deployment Square footage required	Stowage factor
HET	777	0.75
34-Ton truck	498	0.75
5K-Gal POL tanker truck	426	0.75
22-Ton Truck	425	0.75

Deployment Capacity -- DEPCAP$_t$. This coefficient represents the amount of deployment (shipping) square footage available for the support assets modeled. In our base case we assumed 15,000 square feet of shipping arrived daily in theater beginning on day 6 through day 40.

Maximum Asset Availability -- MAX_ASSSET$_{it}$. We held the maximum assets available constant over all time and used the following values:

Asset	Maximum available
HET	768
34-Ton truck	854
5K-Gal POL tanker truck	854
22-Ton Truck	1830

46

Host Nation Support Assets – HNS$_{ti}$ We did not assume the availability of any host nation support assets.

Intra-theater Arc Network Capacity – CAP$_{tr}$ We assumed that each intra-theater arc (road) was capable of carrying 3,000 vehicles per day.

Results

In the remainder of this appendix, we present model results. We chose examples that were relatively simple and yet provided an opportunity to illustrate how the ROSE model could be used to develop options for dealing with a support shortfall. We first present results for a base case that is described in the data structures above. A shortfall is shown to exist for the second unit move in the base case. We then present results for three variations of the base case, each representing a different strategy for addressing the support shortfalls that result in the base case. The first strategy increases the daily lift allotment for support assets from 15k sq ft/day to 20k sq ft/day. The second strategy inserts 50k sq ft of support assets at day 2. The third strategy delays the second and third moves by one week.

Base Case. In the base case we assume 15,000 square feet of shipping arrived daily in theater for carrying support assets[2] (in this simple case, just different types of trucks) beginning on day 6 through day 40, for a total of 450,000 square feet of deployment space. The model both decides which support assets to deploy to theater and allocates the support assets to activities once in theater. In this case the support activities are moving different types of commodities in each of the four unit moves.

Figure B.1 provides an overview of the movement of commodities in the theater (top graph) and the deployment of assets to theater (bottom graph). The movement of commodities in theater are associated with activities. The activities are grouped into four tasks associated with unit moves in theater that are apparent in the top graph. The initial bars in days 6-10 represent the assignment of assets to complete the first unit move. Although the EST for the first move is day 2, work on the task does not begin until day 6, when transportation assets first arrive in theater. Then as assets move into theater they are assigned to work on the first move. Since the move is not

---

[2] Actual shipping may be considerably greater. The 15,000 feet/day represents the difference between total shipping arriving in theater minus the shipping dedicated to shipping combat units and non-modeled support assets (if any).

large, by the end of day 8, assets coming into theater are no longer required for move 1. Hence the LP is moving in assets in anticipation of the future assets (assets are deployed to theater but not assigned to tasks in theater until the beginning of the second unit move).
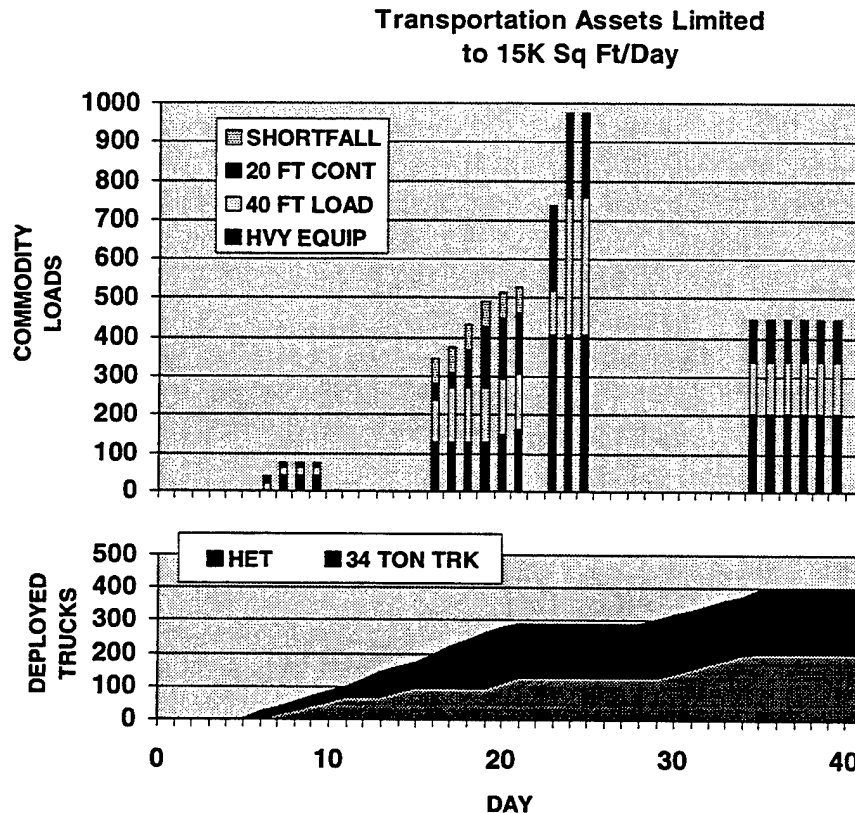
**Transportation Assets Limited
to 15K Sq Ft/Day**

Figure B.1 -- Results for the Base Case (15k sq ft/day)

Assets continue deploying into theater after day 10; then, on day 16, all the current assets in theater are assigned to the second move (the first move has long since been over), which is 168 km long. Assets are also assigned to the second move as they arrive in theater between days 16 and 21 (since they can still provide positive productivity). However, there are not enough assets in theater prior to the LCT for the second move, and a shortfall of 388.63 heavy equipment loads occurs (for graphical purposes the shortfall is spread uniformly across the window associated with the task).

The third unit move is only 21 km long. So although there is an overlap in the window with the second move, no assets are transferred to the third move until the second move's LCT of day 22 has been reached. Because the third move is

48

significantly shorter, implying the assets are more productive, the assets in theater at the end of day 21 are sufficient to complete the third move without a shortfall.

The fourth move is the most taxing one. The commodities to be moved and the time allotted are the same as moves two and three, but the move is 227 km long. Because the TFT weighting factor penalizes the act of using deployment capacity earlier than is necessary, the deployments begin again on day 29 just in time to get all the assets needed for the fourth move on the ECT of day 35. The fourth move requires 204 HETs and 190 34T tractors starting on day 35 and working the entire move window, so once these assets are available in theater, no additional assets are deployed.

The deployment constraint was tight in the base case between days 6 and 21 (15,000 square feet per day), primarily reflecting the demands of move two, which still ended up with a support shortfall. Deployments each of these days varied from 14.48 HETs and no 34-ton trucks to no HETs and 22.59 34-ton trucks (assets deployed and assigned can be non-integer). The constraints associated with maximum assets available and the intra-theater road network were slack each period.

Increase daily support deployments. To address the shortfall in the second unit move, the first strategy is to increase the amount of lift available from 15k sq ft/day to 20k sq ft/day (changes the data for the parameter *DEPCAP*). This could be done by buying or gaining access to more strategic lift assets or a reallocation between combat and support deployments (suggesting some combination of this strategy and the last strategy discussed below, delaying unit moves). Again, the deployment constraints are the only tight constraints, but only to day 18. By the end of day 19, enough assets have been moved into theater to fulfill the demands of move two, which ends on day 21 (for an LCT of 22), without a shortfall (a slack deployment constraint prior to the end of the second move implies the shortfall has been eliminated). From day 6 to day 18, the assets deployed vary between 19.31 HETs and no 34-ton trucks to no HETs to 30.12 34-ton trucks. The final assets are moved in later starting on day 32, just prior to the beginning of the fourth move. Once again the total deployment is 204 HETs and 190 34-ton trucks, the minimum required to complete the fourth move with no shortfall using the entire move window.
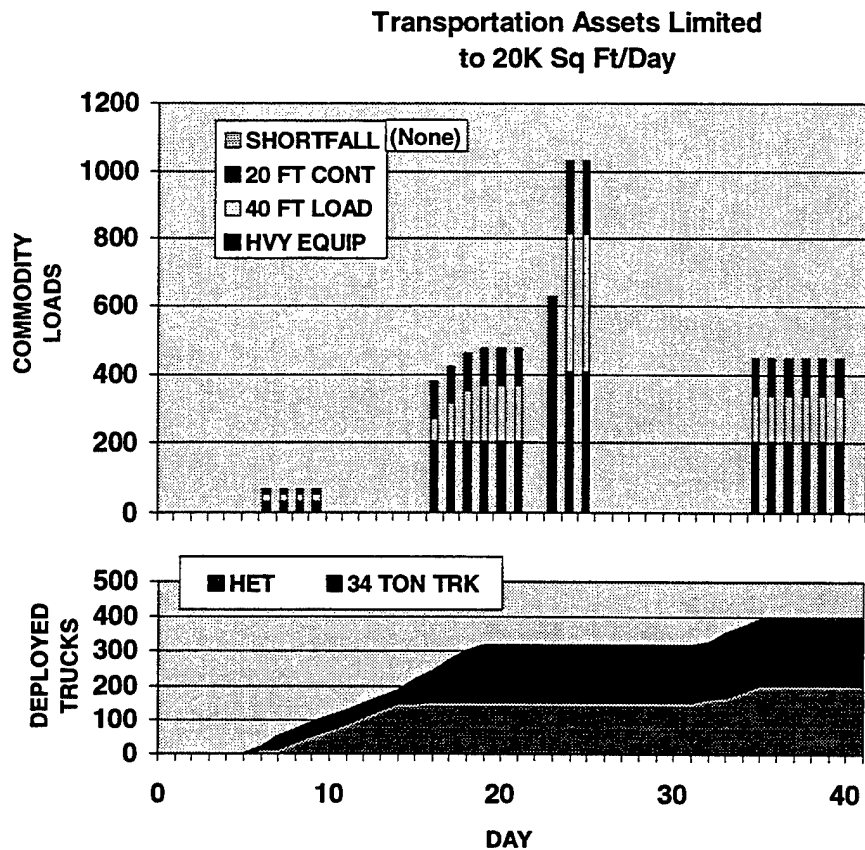
**Transportation Assets Limited
to 20K Sq Ft/Day**

Figure B.2 -- Results for 20k sq ft/day

Use prepositioned assets. To address the shortfall in the second unit move, a
second strategy is to take advantage of prepositioned assets. These assets can be
forward positioned in theater, propositioned on ships, or even host nation support that
is available early on. In this example, we assumed 50,000 square feet of deployment
capacity arriving in theater on the second day (same day as the EST for the first unit
move). This is accomplished by adjusting data for the parameter *DEPCAP*.
Furthermore, we allowed the linear program to decide what support assets to deploy
on the 50,000 square feet.[3]

---

[3] If the prepositioned ships are already configured and cannot therefore be decided by the
model, the specific mix of support assets on the prepo ships can be input to the model.

Transportation Assets Augmented
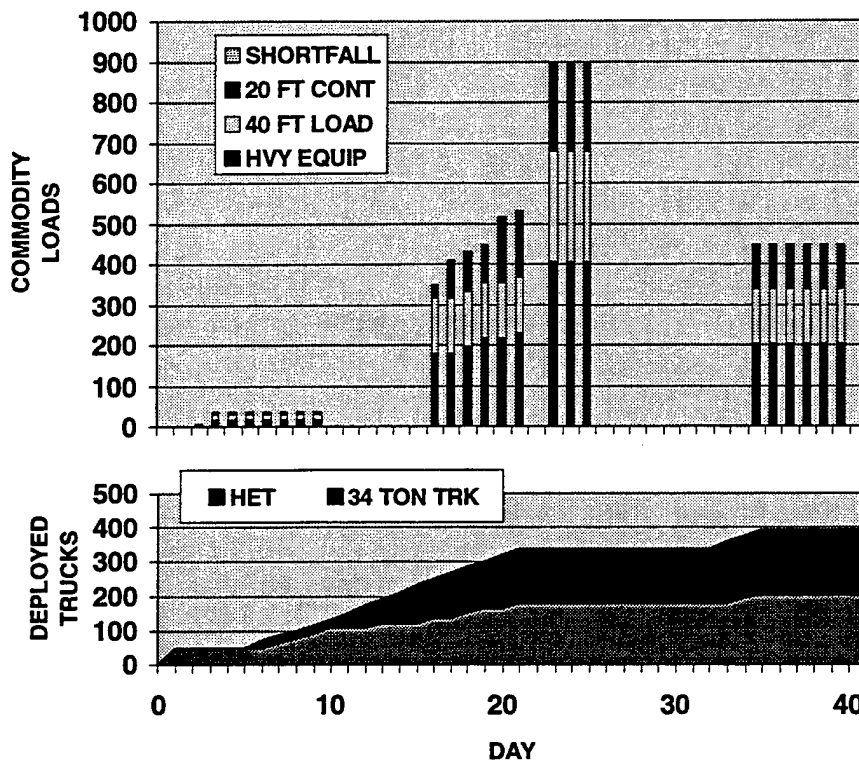## with 50K Sq Ft Prepositioning Arriving Day 3

Figure B.3 -- Results for 15k sq ft/day and 50k sq ft at day 2

The linear program loads the 50,000 sq ft of APS with 48 HETs and two 34-ton trucks, and the deployment constraint for day three is tight. The deployment constraints are also tight for days 6 through 21 as the model attempts to fill the requirements of move 2. Despite the early arriving assets, which are very productive for the second move since they are available for the full move window, there is a slight shortfall of 3 heavy equipment. Deployments are initiated again on day 32 through 35 to satisfy the requirements of the fourth move, and the total deployment is again 204 HETs and 190 34-ton trucks.

Delay division moves. The final strategy to address the shortfall is to delay unit moves 2 and 3 by one week. The EST and LCT parameters for moves 2 and 3 were changed to 23 to 29 and 27 to 33 respectively. The deployment constraints remain tight all the way to day 27, suggesting more assets could have been moved in on day 28 and been productive on move 2. As this suggests, there are no shortfalls for move

51

2. Additional assets are then deployed beginning on day 32 to get the assets in theater (204 HETs and 190 34-ton trucks) required to complete the fourth move without a shortfall.

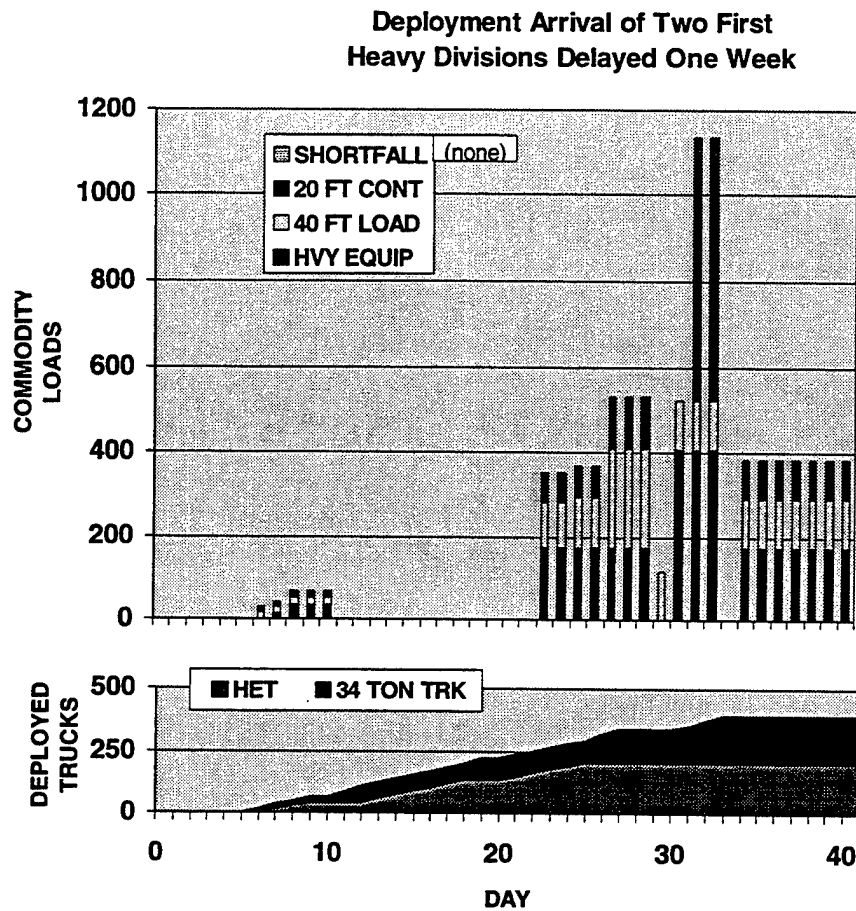**Deployment Arrival of Two First Heavy Divisions Delayed One Week**

Figure B.4 -- Results for 15k sq ft/day and a Delay of Moves 2 and 3