NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

ARES
A SYSTEM FOR REAL-TIME
OPERATIONAL AND TACTICAL DECISION SUPPORT

by
Antonios L. Vassiliou

December 1986

Thesis Advisor: Gerald G. Brown

Approved for public release; distribution is unlimited.
This document is best quality available. The copy furnished to DTIC contained a significant number of pages which do not reproduce legibly.
**REPORT DOCUMENTATION PAGE**

**REPORT SECURITY CLASSIFICATION**
 UNCLASSIFIED

**SECURITY CLASSIFICATION AUTHORITY**

**CLASSIFICATION/DOWNGRADING SCHEDULE**

**REVIEWING ORGANIZATION REPORT NUMBER(S)**

**NAME OF PERFORMING ORGANIZATION**
Naval Postgraduate School

**ADDRESS (City, State, and ZIP Code)**
Monterey, California 93943 - 5000

**NAME OF MONITORING ORGANIZATION**
Naval Postgraduate School

**ADDRESS (City, State, and ZIP Code)**
Monterey, California 93943 - 5000

**NAME OF FUNDING/SPONSORING ORGANIZATION**

**ADDRESS (City, State, and ZIP Code)**

**NAME OF RESPONSIBLE INDIVIDUAL**
Gerald G. Brown

**DATE OF REPORT (Year, Month, Day)**
1986 December

**SOURCE OF FUNDING NUMBERS**

**ABSTRACT SECURITY CLASSIFICATION**
Unclassified

**ABSTRACT**
Introduce a real-time decision support system which uses optimization methods, simulation, and judgement of the decision maker for operational assignment of military units to tasks and for tactical allocation of unit resources to task requirements. The system, named ARES for the Greek god of war, accommodates a high degree of detail in the logistics of unit movements and operations, yet separates the assignment and allocation activities in a fashion which naturally accommodates human intervention and judgement. It is designed to assist the decision maker, not to replace him. ARES is demonstrated in a hypothetical scenario constructed for Engineering Battalions of the Hellenic

**DISTRIBUTION/AVAILABILITY OF ABSTRACT**

**DISTRIBUTION/AVAILABILITY OF REPORT**
Approved for public release; distribution is unlimited

**DISTRIBUTION/AVAILABILITY OF THIS PAGE**

**COSATI CODES**

**SUBJECT TERMS**

**PROJECT OFFICE SYMBOL**
54

**OFFICE SYMBOL**
408-646-2140

**TELEPHONE**
55BW

**PAGE COUNT**
44
Army which are to be assigned 20 tasks employing 25 resource types in repairing major damage to public works following a large earthquake*. ARES is designed for use in real time, and quick data preparation is aided by the provision of standard task icons from published sources.

* This hypothetical data was prepared prior to the earthquake in Kalamata near Athens on 13 September, 1986, and exhibits uncanny, but coincidental resemblance to that real situation.
ARES
A System for Real-Time Operational and Tactical Decision Support

by
Antonios L. Vassiliou
Major, Hellenic Army
B.S., Military Academy of Greece, 1974

Submitted in partial fulfillment of the requirements for the degree of
MASTER OF SCIENCE IN MANAGEMENT
from the
NAVAL POSTGRADUATE SCHOOL
December 1986

Author:
Antonios L. Vassiliou

Approved by:
Gerald G. Brown, Thesis Advisor
Benjamin J. Roberts, Second Reader
Willis R. Greer Jr., Chairman, Department of Administrative Sciences

Kneale T. Marshall, Dean of Information and Policy Sciences
ABSTRACT

We introduce a real-time decision support system which uses optimization methods, simulation, and judgement of the decision maker for operational assignment of military units to tasks and for tactical allocation of unit resources to task requirements.

The system, named ARES for the Greek god of war, accommodates a high degree of detail in the logistics of unit movements during operations, yet separates the assignment and allocation activities in a fashion which naturally accommodates human intervention and judgement.

ARES is designed to assist the decision maker, not to replace him. ARES is demonstrated with a hypothetical scenario constructed for 14 Engineering Battalions of the Hellenic Army which are assigned 20 tasks employing 25 resource types in repairing major damage to public works following a great earthquake*.

ARES is designed for use in real time, and quick data preparation is aided by the provision of standard task icons from published sources.

* This hypothetical data was prepared prior to the earthquake in Kalamata near Athens on 13 September, 1986, and exhibits uncanny, but coincidental resemblance to that real situation.
# TABLE OF CONTENTS

I. INTRODUCTION ....................................... 7

II. OPERATIONAL ASSIGNMENT MODEL (IP) ............ 12

III. OPERATIONAL ASSIGNMENT MODEL (IPL) .......... 15

IV. TACTICAL ALLOCATION MODEL (GNK) ............... 18

V. CONSIDERATION OF LOGISTICS ....................... 21

VI. AN EXAMPLE SCENARIO ................................ 23

VII. DESIGN AND IMPLEMENTATION ....................... 28

VIII. SCENARIO RESULTS ................................ 32

IX. COMPUTATIONAL EXPERIENCE .......................... 35

X. DISCUSSION AND CONCLUSION ......................... 38

LIST OF REFERENCES ..................................... 41

INITIAL DISTRIBUTION LIST ............................. 43
ACKNOWLEDGEMENT

First of all I would like to express my most sincere appreciation and thanks to Professor Gerald G. Brown, my thesis advisor, with whose support, professional assistance, and indefatigable contribution and advise, this 'ARES' system has been successfully completed.

Also I would like to thank professor Benjamin J. Roberts, the second reader, for his suggestions and comments.

I am deeply indebted to my wife, Paraskeyi, and two sons, Constantine and Alexander, for their patience, understanding, and assistance.

Lastly I express my love to my newborn daughter Veronica whose cries inspired me to work harder during the long nights devoted to the study of this "ARES" system.
I. INTRODUCTION

We introduce ARES, a prototypic system for real-time operational and tactical decision support. ARES is designed to quickly and effectively help respond to complex emergent problems in disaster relief, in the operational art and tactics of warfare, and in related multiperiod, large-scale employment of heterogeneous, substitutable resources restricted in availability and demand over time, over geography, and by organizational limitations.

Although a great deal of work has been done in strategic modelling in many contexts, there is relatively little available modelling help beyond simple thumb rules for the time-pressed (operational or tactical) decision maker to translate strategic goals into logistically constrained operational and tactical plans (and the issues are different). The luxuries of hypothetical additional resources and the time to analyze their employment are just not available in the operational and tactical domains: operational and tactical decisions must be made quickly, and usually involve employing only resources actually available to perform whatever mission is at hand.

The history of assignment and allocation models for planning emergency logistics extends back to some of the earliest work in linear programming, game theory, and their economic interpretation. We cite only a few of the references in this large body of literature. The seminal works by Dantzig and by Koopmans (both found in Koopmans 1951) are explicitly motivated by large-scale logistics problems. Karchere and Hoeber (1953) give early direction on the use of newly developed optimization technology in weapon
system planning and allocation, discussing substitutability of resources and choice of suitable objective functions. Geisler (1959) reports RAND's first use of man-machine simulation of logistics support activities.

Chaiken and Larson (1972) state some basic issues in logistic location and task assignment for emergency service vehicles: how many units should there be, where should they be located, who should they serve, and how can they be relocated to substitute for units not available. Kaplan (1973) redeploy divisible resources with linear programming. Fitzsimmons (1973) states a nonlinear response time model and uses pattern search to locate units well and allocate workload equitably. Swoveland, Uyeno, Vertinsky, and Vickson (1973) employ simulation and human interaction to set up a unit location problem as a quadratic assignment model which is solved with an elegant heuristic. Bracken and McGill (1974) formulate strategic force planning models as two-sided games solved with nonlinear programming. Bracken, Falk, and Karr (1975) apply multiperiod, two-person zero-sum games formulated to develop strategies for unit sortie allocations.

Finally, Kolesar and Walker (1974) develop a multi-stage solution approach to unit and task assignment using set covering and transportation-like integer linear programs which are used in real time by applying heuristics.

Named for the Greek god of war, ARES is a proof prototype of a real-time decision support system. It employs optimization and simulation to capture and exploit a high degree of realism without demanding unreasonable amounts of data, or locking the decision maker out of the decision process. The intent is to provide quick credible advice with good global perspective at a cost no greater than the relatively myopic
decision methods now widely used.

ARES accommodates enough detail to support realistic decisions, but not so much as to render the process useless. For the intended applications, the particular missions to be performed will not likely be known much in advance, but the generic types of missions are known and can be planned. ARES uses a taxonomy of prepared standardized icons for data describing possible missions. The idea is to help the decision maker quickly assemble a data scenario closely resembling the proximate situation from a menu of these standard icons.

We characterize the mission at hand as a set of geographically dispersed tasks, each composed of partially-ordered sub-tasks requiring over time varying amounts of different resources. Organizational units, also geographically dispersed and each possessing a different endowment of resources, are to be assigned responsibility for the tasks. Responsibility for each task rests with only one unit at any given time.

ARES consists of several models coordinated by a time interval simulator which also scales and manipulates scenario data in a fashion transparent to the decision maker. Two integer linear programs, a linear program, a georeference system, a mobility system, a decision-maker simulator, and extensive user interface and user override and control facilities complete the program suite. The models in ARES all use a standard data interface visible to the decision maker; this invites expansion with new models and features.

A scenario is created from the attributes of available units and task attributes derived in large part from standard cataloged data icons for similar tasks. A georeference system is accommodated to generate distance
costs and delays in relocating and operating units. The decision maker may preview the scenario and modify data or manually pre-assign tasks and units as he sees fit.

Operational assignment of tasks to units uses one of two integer programming models (IP) or (IP). Good task aggregations for the unit assigned reduce unit relocation costs and match unit resource endowments with aggregated task resource requirements. Logistical considerations are paramount at this stage.

The decision maker can review the operational assignments, modify them manually, or reject them outright and restate the conditions for the original operational assignment scenario. An acceptable set of operational assignments is passed forward to a tactical model.

Tactical allocation of the resources of each unit to the requirements of its assigned tasks uses a linear programming model (GN). Substitutions among resources are permitted, although at reduced efficiencies in completing the tasks. Allocations recognize task priorities and the logistical effects of geographic proximity. In addition, unit efficiency in performing a particular task improves over time, and the sequence within tasks of resource requirements is considered. The result is a complete plan for each unit, showing what resources are to be used to fulfill each task requirement, and the efficiency with which operations are expected to be carried out. The allocation also determines which requirements will not be met in situations which overtax units.

Finally, the decision maker is presented with a complete solution, which he can accept, or modify, or reject outright and repeat. Regardless of his action, ARES is designed to lend quick insight. The decision maker
uses his own judgement concerning non-quantified factors, and he should gain a deeper understanding of the situation at hand from ARES.
II. OPERATIONAL ASSIGNMENT MODEL (IP)

This integer program finds good aggregate assignments of tasks to units without explicit consideration of unit movement.

Index Use

\( i \) Tasks

\( j \) Resources

\( k \) Units

Given Data

\( d_{ik} \) Distance cost from unit \( k \) to task \( i \)

\( r_{ij} \) Minimum, maximum resource \( j \) requirements of task \( i \)

\( a_{jk}, \bar{a}_{jk} \) Minimum, maximum resource \( j \) employable by unit \( k \)

\( z_{jk}, \bar{z}_{jk} \) Penalties for violating minimum, maximum resource limits

\( p_i \) Priority of task \( i \) (>0)

\( u_i, \bar{u}_i \) Penalties for not assigning, double-assigning task \( i \)

\( f_{jj} \) Substitution efficiency of resource \( j \) (>0)

\( s_{ij} \) Sequence of resource \( j \) requirement within task \( i \)

\( h_{ijk} \) Consumption by task \( i \) of resource \( j \) from unit \( k \)

Decision Variables

\( x_{ik} \) Binary variable for assigning task \( i \) to unit \( k \)

Formulation

\[
\text{MIN } \sum_{ik} d_{ik} x_{ik}
\]

s.t. \[ \sum_{k} x_{ik} \geq (1,1) : (u_i, \bar{u}_i) \text{ for all } i \] \( (1) \) (GUB)

\[ \sum_{i} h_{ijk} x_{ik} \geq (a_{jk}, \bar{a}_{jk}) : (z_{jk}, \bar{z}_{jk}) \text{ for all } j,k \] \( (2) \)

\[ x_{ik} = \{0,1\} \text{ for all } i,k \] \( (3) \) (IP)
The notation \( o (r, \overline{r}; (z, \overline{z}) \) indicates lower and upper ranges \( (r, \overline{r}) \) on row functional values with corresponding respective linear penalties per unit of violation \( (z, \overline{z}) \); i.e., this is a goal program with linear penalties, an elastic integer program (Brown and Graves 1975).

Constraints (1) encourage assignment of each task to some unit, and form a generalized upper bound (GUB) row set (Dantzig and Van Slyke 1967).

Constraints (2) express the goodness of fit of task assignments with employable unit resources. Constraints (3) preclude fractional assignment of tasks to units.

Consumption by task \( i \) of resource \( j \) from unit \( k \) is defined:

\[
h_{ijk} \equiv r_{ij} e^{-\ln f_{jj} + (p_i - 1)/10 + d_{ik}/\sigma_k + t_{ik}^2}.
\]

where \( \sigma_k \) is the speed of advance of a unit and \( t_{ik} \) is the number of periods that unit \( k \) has already been assigned task \( i \). The rationale for the particular consumption function (1) amplifies the resource requirement \( r_{ij} \) to account for the state of resource readiness \( f_{jj} \), the task priority \( p_i \) (making less important tasks appear more expensive), the logistic proximity of unit \( k \) and task \( i \), \( d_{ik}/\sigma_k \), and learning curve effect as a function of time since assignment, \( t_{ik} \). The data are scaled so that (1) is in conformity with policy guidance or the judgement of the decision maker. Alternate consumption functions may appeal in other situations.

The distance costs \( d_{ik} \) and penalties \( u_i \). \( \overline{u}_i \) and \( z_{jk} \). \( \overline{z}_{jk} \) are expressed in commensurate units and deserve some thought by the modeller. For instance, \( \overline{z}_{jk} \) may be interpreted as how much additional distance cost should be incurred before considering overtaxing maximum resource.
employment $a_{jk}$ for unit $k$; this is a direct expression of logistical efficiency. For simplicity in our tests, distance costs $d_{ik}$ are scaled by a policy parameter. $z_{jk}$ and $\bar{z}_{jk}$ are part of the input script. $u_i$ is defined as $100/p_i$, and $\bar{u}_i$ equals 100.
III. OPERATIONAL ASSIGNMENT MODEL (IP_L)

The purpose of this integer program is to find good movements of units to locations from which they will be assigned good aggregate groups of tasks to perform.

Index Use

- $i$: Tasks
- $j$: Resources
- $k$: Units
- $l$: Locations (assumed here to be collocated with tasks)

Given Data

- $d_{lk}$: Distance cost from unit $k$ to location $l$
- $g_{il}$: Distance cost from task $i$ to location $l$
- $r_{jli}$: Gross resource requirement $j$ of task $i$ performed from location $l$
- $a_{jlk}$: Net resource availability $j$ of unit $k$ located at $l$

Decision Variables

- $z_{il}$: Binary variable for assigning task $i$ to location $l$
- $x_{lk}$: Binary variable for moving unit $k$ to location $l$
Formulation

\[
\text{MIN } \sum_{i} z_{i1} + \sum_{kl} d_{lk} x_{lk} \quad \text{for all } i \quad (1) \quad (\text{GUB})
\]

\[
\sum_{i} z_{i1} = (1.1);(u_{i1},\bar{u}_{i1}) \quad \text{for all } i \quad (2) \quad (\text{GUB})
\]

\[
\sum_{k} x_{lk} = (1.1);(m,m) \quad \text{for all } k \quad (3)
\]

\[
-x_{i1} + \sum_{k} x_{lk} = (0.1);(m,m) \quad \text{for all } i \quad (4)
\]

\[
-\sum_{j} \tilde{r}_{j1} z_{i1} + \sum_{kl} \tilde{a}_{jlk} x_{lk} = (0.0);(b,b) \quad \text{for all } i,j \quad (5)
\]

\[
z_{i1} = (0.1) \quad \text{for all } i \quad (6)
\]

\[
x_{lk} = (0.1) \quad \text{for all } i,k \quad (7) \quad (\text{IP}_{L})
\]

\(\text{(IP}_{L}\) uses the notation of (IP). Constraints (1) encourage assignment of each task to some location. Constraints (2) allow movement of each unit to some location. (A GUB row set is formed by constraints (1) and (2).) Constraints (3) attempt to restrict assignments so at most one unit is moved to any particular location, (4) require that a unit be moved to any location to which a task is assigned, and (5) attempt to match for each location and each resource an aggregate assignment of tasks which have gross resource requirements about equal to the net resource availability of the unit moved to that location to perform the tasks. Constraints (6) and (7) preclude fractional location of tasks and units.

Gross resource requirement \(\tilde{r}_{j1}\) represents the resource \(j\) estimated to be required at location \(1\) in order that task \(i\) actually receive \(r_{ij}\).

\[
\tilde{r}_{j1} = r_{ij} e^{-\ln f_{jj} + (p_{i1})/10 + z_{i1}/\sigma} \quad (2)
\]

where \(\sigma\) expresses the logistic radius of influence from any location; we
have used $\sigma = 100$. The gross resource requirement (2) amplifies the resource requirement $r_{ij}$ in the same fashion as (1).

Net resource availability $\tilde{a}_{jlk}$ represents the amount of resource $j$ which unit $k$ can deliver from its endowment $\overline{a}_{jk}$ forward to location $l$. Unit $k$ may be moving toward location $l$ while supplying this net resource.

$$\tilde{a}_{jlk} \equiv \overline{a}_{jk} f_{jj} \left( \alpha_{lk} + (1-\alpha_{lk}) e^{-d_{lk}/\sigma_k} \right).$$

(3)

where $\alpha_{lk}$ is the fraction of time which the unit will spend at its destination location, and $\sigma_k$ is the speed of advance.

The distance costs $d_{lk}$ and $g_{il}$, and the penalties $u_i$, $\overline{u}_i$, $m$, $b$ and $\overline{b}$ all render the same objective function units. In our work, $m \equiv 100$, and $u_i$ and $\overline{u}_i$ are defined as in (IP). The penalties for assigning too little (or too much) resource $j$ to location $l$ are $b$ (or $\overline{b}$). We have used $b = 0.1$ and $\overline{b} = 0.01$. 

IV. TACTICAL ALLOCATION MODEL (GNₖ)

This linear program allocates resources to the tasks assigned to unit k.

Index Use

i  Tasks
j  Resources
w  Work (resources required by tasks assigned to unit k)

Given Data

\( r_{iw}, \bar{r}_{iw} \)  Minimum, maximum work requirements w of assigned task i
\( q_{iw}, \bar{q}_{iw} \)  Penalties for violating minimum, maximum work requirements
\( a_{jk}, \bar{a}_{jk} \)  Minimum, maximum resource j employable by unit k
\( Z_{jk}, \bar{Z}_{jk} \)  Penalties for violating minimum, maximum resource limits
\( p_i \)  Priority of task i
\( f_{jw} \)  Substitution efficiency of resource j for work requirement w (>0)
\( s_{iw} \)  Sequence of work requirement w in task i (>0)
\( e_{iwj} \)  Efficiency of resource j used for work w on task i

Decision Variables

\( y_{iwj} \)  Allocation of resource j to task i resulting in work w

Formulation

MAX \sum_{iw} e_{iwj} y_{iwj}

st \sum_{j} y_{iwj} = (r_{iw}, \bar{r}_{iw}) : (q_{iw}, \bar{q}_{iw}) \text{ for all } i, w \quad (1) \text{ (GUB)}
\sum_{iw} e_{iwj} y_{iwj} = (a_{jk}, \bar{a}_{jk}) : (Z_{jk}, \bar{Z}_{jk}) \text{ for all } j \quad (2)
\sum_{iw} \sum_{j} y_{iwj} = 0 \quad \text{ for all } i, w, j \quad (3) \text{ (GNₖ)}
(GN<sub>k</sub>) uses the notation of (IP). However, the dimensions of (GN<sub>k</sub>) discriminate between resources consumed, j, and work completed, w, explicitly representing substitutability of resources. Constraints (1) encourage allocation of sufficient work resources, while constraints (2) indicate the desired mix of employable unit resources. Constraints (3) require non-negative resource allocations. (GN<sub>k</sub>) is an elastic generalized network (Brown and McBride 1984).

Efficiency of resource j used for work w on task i is defined

\[ e_{iwj} = e^{-ln f_{jw} + (p_{i}-1)/10 + s_{iw}^{+}/10 + \bar{d}_{ik}/\sigma_{k}}. \]  

(4)

where \( s_{iw}^{+} = \max \{0, s_{iw} - t\}\), t is the last time period of this allocation, and \( \sigma_{k} \) is the unit speed of advance. The efficiency (4) employs the readiness and substitutability of resources via \( f_{jw} \cdot s_{iw}^{+} \), reduces efficiency if the work w should not be started until period \( s_{iw}^{+} \).
If model (IP) has been used for operational assignment.

\[ \overline{d}_{ik} \equiv \max \{0, d_{ik} - \sigma_k/2\}. \] (5)

If unit \( k \) is to be advanced toward, or to, location \( l \) by model (IP_L).

\[ \overline{d}_{ik} \equiv \max \{0, d_{ik} - \sigma_k/2\} + g_{il}. \] (6)

These distance costs \( \overline{d}_{ik} \) in (5) or (6), and penalties \( g_{iw}, \overline{q}_{iw}, \overline{Z}_{jk}, \) and \( \overline{Z}_{jk} \) are all intended to yield the same objective function units. For our tests, \( g_{iw} \equiv 100/p_i s^+_{iw} \) and \( \overline{q}_{iw} \equiv 100. \)
V. CONSIDERATION OF LOGISTICS

The efficiency with which a unit completes a task depends heavily upon logistical considerations. If a unit is remote from a task, or must be moved, its efficiency suffers. Figure 1 shows an idealized situation with unit k, task i, and location l.

![Diagram of idealized geographic logistical scenario]

Figure 1: Idealized Geographic Logistical Scenario

Model (IP) assigns tasks to units relying exclusively upon $d_{ik}$. (IP) moves units to new locations and assigns tasks to be performed from these new unit locations. (IP) recognizes $d_{ik}$ and $g_{il}$. The distances $d_{ik}$ and $d_{ik}$ are surrogates for logistical costs of assignment during the ensuing time period. Clearly, (IP) is more appropriate for situations in which unit movements are expected. (IP) when they are not. (PL) provides the decision maker with a better opening gambit than does (IP) if the scenario involves significant initial redeployment of units.

Tactical allocation models (ON) are given unit and task assignments and planned unit movements. Therefore, (ON) can allocate resources using...
any logistic efficiency function of assigned distances, and of other attributes induced only from assignment such as weather effects, speed of unit movement, etc. (GN) can also substitute resources at somewhat reduced efficiency as well as prioritizing their immediate application. Given a fairly reasonable operational assignment, (GN) provides a high-resolution work plan with rich logistic detail and good face validity.
VI. AN EXAMPLE SCENARIO

We demonstrate ARES with an example constructed for Engineering Battalions of the Hellenic Army. The mission scenario involves 20 tasks repairing major damage to public works following an earthquake. For our purposes, there are 14 units, each endowed with some of 25 resources. Figure 2 shows the units and tasks from the ARES input script. In the United States, the Department of the Army defines unit types in (1976), and task standards in (1973a).

<table>
<thead>
<tr>
<th>UNIT LABELS, LOCATIONS, AND PRIOR ASSIGNMENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>UU</td>
</tr>
<tr>
<td>----</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>5</td>
</tr>
<tr>
<td>6</td>
</tr>
<tr>
<td>7</td>
</tr>
<tr>
<td>8</td>
</tr>
<tr>
<td>9</td>
</tr>
<tr>
<td>10</td>
</tr>
<tr>
<td>11</td>
</tr>
<tr>
<td>12</td>
</tr>
<tr>
<td>13</td>
</tr>
<tr>
<td>14</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TASK LABELS, LOCATIONS, PRIORITIES, AND PRIOR ASSIGNMENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>TT</td>
</tr>
<tr>
<td>----</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>5</td>
</tr>
<tr>
<td>6</td>
</tr>
<tr>
<td>7</td>
</tr>
<tr>
<td>8</td>
</tr>
<tr>
<td>9</td>
</tr>
<tr>
<td>10</td>
</tr>
<tr>
<td>11</td>
</tr>
<tr>
<td>12</td>
</tr>
<tr>
<td>13</td>
</tr>
<tr>
<td>14</td>
</tr>
<tr>
<td>15</td>
</tr>
<tr>
<td>16</td>
</tr>
<tr>
<td>17</td>
</tr>
<tr>
<td>18</td>
</tr>
<tr>
<td>19</td>
</tr>
<tr>
<td>20</td>
</tr>
</tbody>
</table>

Figure 2: Units and Tasks of Example
The georeference coordinates of units and tasks are given in Figure 1 for the situation depicted in Figures 3, 4 and 5.

Figure 3: Initial Geographic Locations of Units. (Coordinates displayed are a georeference in common with the following figures.)
Figure 4: Earthquake Epicenter.

Figure 5: Geographic Locations of Tasks. Geographic locations of damaged public works and earthquake epicenter are shown.
A georeference system is used to generate coordinate-to-coordinate distance costs, which appear in the ARES input script.

The resource requirements for Task 1 ("ADMIN. BUILDING AA1051"), a disaster relief facility, are given in Figure 6 and the input script. Resource requirements such as these are available in standard engineering reference manuals for a wide variety of task types (for instance, see unclassified sources from the United States Department of the Army (1973a, 1973b, and 1973c). We envision a taxonomy of standardized task data icons from which a particular set of requirements can be very quickly extracted and assembled for a scenario. The size of our resource requirements data base is modest but the resulting accuracy and level of detail are quite good. Better yet, data mobilization from a menu of such icons can be completed in minutes.

<table>
<thead>
<tr>
<th>RESOURCE LABELS AND TASK 1 REQUIREMENTS</th>
<th>MAN</th>
</tr>
</thead>
<tbody>
<tr>
<td>RR I LABEL</td>
<td>I HOURS</td>
</tr>
<tr>
<td>1 ENGINEER-PION-APREN-MLPER</td>
<td>4448</td>
</tr>
<tr>
<td>2 SURVEYOR</td>
<td>70</td>
</tr>
<tr>
<td>3 CARPENTER</td>
<td>7557</td>
</tr>
<tr>
<td>4 ELECTRICIAN</td>
<td>948</td>
</tr>
<tr>
<td>5 PLUMBER</td>
<td>1740</td>
</tr>
<tr>
<td>6 MASON</td>
<td>1420</td>
</tr>
<tr>
<td>7 STRUCTURE SPECIAL.</td>
<td>0</td>
</tr>
<tr>
<td>8 HEAT-VENTILAT SPECIAL.</td>
<td>200</td>
</tr>
<tr>
<td>9 WELDER</td>
<td>0</td>
</tr>
<tr>
<td>10 PIPELINE</td>
<td>0</td>
</tr>
<tr>
<td>11 CRANE-SHOVEL OPER.</td>
<td>200</td>
</tr>
<tr>
<td>12 LOADER OPER.</td>
<td>150</td>
</tr>
<tr>
<td>13 DOZER OPER.</td>
<td>100</td>
</tr>
<tr>
<td>14 COMPRESSOR OPER.</td>
<td>0</td>
</tr>
<tr>
<td>15 DUMP TRUCK OPER.</td>
<td>400</td>
</tr>
<tr>
<td>16 CONCRETE MACHINE OPER.</td>
<td>0</td>
</tr>
<tr>
<td>17 GRADER OPER.</td>
<td>529</td>
</tr>
<tr>
<td>18 CRUSHER OPER.</td>
<td>0</td>
</tr>
<tr>
<td>19 DITCH MACHINE OPER.</td>
<td>570</td>
</tr>
<tr>
<td>20 ASPHALT SPECIAL.</td>
<td>0</td>
</tr>
<tr>
<td>21 POWER ROLLER OPER.</td>
<td>0</td>
</tr>
<tr>
<td>22 WATER DISTRIBUT. OPER.</td>
<td>0</td>
</tr>
<tr>
<td>23 POWER BAT OPER.</td>
<td>0</td>
</tr>
<tr>
<td>24 ROTARY TILLER OPER.</td>
<td>0</td>
</tr>
<tr>
<td>25 SCRAPER OPER.</td>
<td>0</td>
</tr>
</tbody>
</table>

Figure 6: Resource Requirements of Task 1
The resources employable by Unit 1 ("1ST COMBAT BN"), a combat engineering battalion, are given in Figure 7 and in the input script. These resource endowments are in line with those given by the United States Department of the Army (1973a) with conversion to man hours from (1971).

The input script also includes for each task the sequence of resource requirements expressed as the first period when the resource is best applied, and for each resource its substitution efficiency for other resources.

The scenario data constitutes about 1,000 records. However, these records derive from the unit, task, and resource definitions which are modest in number.

Figure 7: Resource Endowment of Unit 1.
VII. DESIGN AND IMPLEMENTATION

ARoS is intended to help the decision maker, not to replace him. Figure 8 shows the functional structure of ARES. The design is biased toward interactive use with review and intervention options at each stage of operational assignment and tactical allocation.
INITIALIZE: Define NEW_SCRIPT

NEXT_PERIOD: Redefine NEW_SCRIPT as OLD_SCRIPT

OP_ASSIGN: Select Model (IP_L) or (IP)

Read OLD_SCRIPT

Generate and Solve (IP_L) or (IP)

Record task and unit assignments on ASSIGN_FILE

REVIEW_IP: Option to review assignments in ASSIGN_FILE

either stop,
or edit OLD_SCRIPT and GOTO OP_ASSIGN,
or edit OLD_SCRIPT and/or ASSIGN_FILE and continue

TAC_ALLOC: Read OLD_SCRIPT and store as SCRIPT

Read ASSIGN_FILE and update SCRIPT assignments

UNIT-K: Select (CN_k). Generate and Solve

Update SCRIPT resource requirements for work completed

For next unit k REPEAT UNIT-K

NEW_SCRIPT: Update SCRIPT unit locations and distance costs

Write SCRIPT as NEW_SCRIPT

REVIEW_PERIOD: Option to review results

either stop,
or edit OLD_SCRIPT and/or ASSIGN_FILE

and GOTO OP_ASSIGN

or edit NEW_SCRIPT and GOTO NEXT_PERIOD

Figure 8: ARES Functional Specification
ARES is implemented in FORTRAN H(Extended) and executes on an IBM 3033 AP computer using the VM/CMS operating system. Input scripts are read from files which may be viewed and modified with a full-screen editor such as XEDIT. (Software copyrights IBM Corporation.)

ARES uses the X-SYSTEM (Brown and Graves 1975) to solve (IP_\text{L}), (IP) and (GN_k) in real time. For each problem instance, problem generators directly convert input script data into an internal representation, the solver is invoked, and the solution is provided to a report writing program. ARES consists of a set of open subroutines and is executed with whatever preview, review, or other external interference is deemed desirable.

We envision cyclic use and review at varying levels of detail as a mission progresses over time. Accordingly, input scripts include the beginning period and number of periods in the ensuing time interval, which intrinsically scales time-dependent input data to the desired level of aggregation. We have tested ARES manually and by replacing the decision maker with a simulation which performs "judgement review" of successive solutions over time. This permits totally automatic evaluation of complete mission scenarios, and avoids tedious manual effort in our research. (A single time interval may generate 15, or 20 thousand lines of solution detail at the scale of our example scenario.)

The update of unit coordinate locations and distance costs is a simple surrogate for a more realistic and complicated georeference and mobility system. ARES estimates the direction and speed of advance of each unit during the time interval and relocates the unit. Then the distance costs are adjusted. If operating areas are known sufficiently in advance to
permit preparation of detailed georeference and mobility systems. ARES can
accommodate the increased level of detail in real time (e.g., Brown, Ellis,
Graves, and Ronen 1987). The update can also be used to degrade, or to
amplify unit resource endowments and effectiveness to modify task resource
requirements, or to change any other data artifact, providing a rich
modelling arena.
ARES has been used in simulation mode to completely plan mission scenarios from start to finish. For the earthquake scenario, Figure 9 shows the initial operational assignments of $(IP_L)$. 

Figure 9: Initial Operational Assignments of Units. Directional vectors show the straight-line path and relative speed of advance $\sigma_k$. 

32
Figure 10 depicts the arrival of units to their initially assigned locations.

Figure 10: Initial Operational Assignments of Units to Locations. Arrows show straight-line path of advance toward assigned locations.

Without intervention by the decision maker, the scenario completed itself in 7 weekly intervals, requiring less than 2 minutes in a 1.2 megabyte memory region.

Face validity of the simulation solution is excellent. No manual intervention has been found to improve the solution. In fact, many manual attempts to coerce better assignments resulted in startling degradations.
The application of available resources, with allowable substitutions, is shown in Figure 11 for the 7 single-period time intervals required to complete the earthquake scenario.

**Figure 11:** Resource Requirements and Work Completed. Each row represents a resource requirement over time-interval columns. The white bars depict resource requirements by time interval; the black bars show the relative fulfillment of the requirements. Broken bars are out of scale. From each time interval to the next the requirements are reduced by the work completed and amplified by new sequence-dependent requirements. In this scenario, 7 weekly time intervals are required to complete all tasks.
IX. COMPUTATIONAL EXPERIENCE

Extensive computational experience reveals that the operational assignment models (IP) and especially (IP_L) are most difficult to solve at the beginning of a scenario, and get progressively easy in later time intervals. The size of these models varies with the number of mandated assignments, impossible assignments, and the non-zero density of resource availabilities and remaining requirements. (IP) typically has about 340 constraints, 268 binary variables, and 6,200 non-zero consumption coefficients. The linear program continuous relaxation can be generated and solved in about 5 seconds, and an optimal binary solution is achieved in another second, or so.

(IP_L) has about 1,000 constraints, 645 binary variables, and 8,000 rather unwieldy non-zero gross resource requirement and net resource availability coefficients.


Briefly, the rows of constraints and columns of variables are lexicographically sorted to place short rows first accompanied by other rows and columns with intersecting non-zero coefficients, and longer rows later with their own intersecting rows and columns.

The problem cascade proceeds by activating a set of constraints, relaxing all other constraints, and activating a set of variables, fixing
all other variables to their last-known values. This problem is solved, the new values of the active variables recorded, and another problem specified in the building problem cascade. The last problem in the cascade activates all constraints and variables (precisely the problem found intractable above) and solves it by starting with an advanced solution recorded from the last-known values of variables solving previous problems in the cascade.

(IPL) resisted even the problem cascade until a new heuristic cascade strategy was adopted which activates the shortest 1/2 of constraints and their associated variables, then the shortest 3/4, then 7/8, and so forth until the last constraint is added and the problem is solved. Remarkably, this approach has been absolutely reliable and robust, while most others fail or prove unruly.

Generation and complete problem cascade solution of the continuous relaxation of (IP_L) now requires about 10 seconds.

An acceptable binary solution to (IP_L) is achieved in another second, or two.

We do not routinely seek optimal binary solutions to (IP_L), which we refer to as "perfect misfits". The gross resource requirements and net resource availabilities in (IP_L) are rough logistic estimates, calibrated by actual field experience but ultimately just approximate target performance levels. For interesting operational assignments (i.e., early in the scenario) there are simply no feasible solutions; the goal is to guess where to send units so that they can peremptorily cope with the mission at hand with maximal effectiveness. Accordingly, we accept in practice binary solutions which may be as much as 25% greater than an
optimal lower bound in total value, including constraint violation penalties. Experimentally, we have determined at additional computational cost (as much as 5 minutes per trial) that these binary solutions are actually almost always within a few percent of the true optimum.

A decision maker can help ARES with its operational assignments or completely specify a solution with manual assignment features. Our experience suggests that the decision maker can express some non-quantifiable guidance in this fashion, but can not hope to apply a remotely competitive global perspective. Manual competition with ARES reveals that model computation effort is amply justified by the quality of operational assignments achieved. The operational assignment models, especially \((I_P_L)\), produce solutions no decision maker is likely to discover. Some of these solutions have yielded remarkable insights. The initial operational commitment of units is arduous and crucial to mission success. \((I_P_L)\) is worth the computational investment.

By contrast, the tactical allocation models \((G_N)\) are easy to solve even in the cases where heroic substitution of resources are required. The size of each \((G_N_k)\) varies with the number of tasks assigned to the unit, and the non-zero densities of resource availabilities, remaining requirements, and allowable substitutions. For our scenario, a typical instance of \((G_N_k)\) has about 70 constraints and 1,190 variables, and is generated and solved in less than 0.1 second. Stress tests with 525 constraints and 12,500 variables require less than a second.
DISCUSSION AND CONCLUSION

The subtlety of operational assignment has surprised us, as has the ease of detailed tactical allocation. Operational assignments are delicate decisions, and the success of entire missions appears to be very sensitive to minute details---precisely the considerations a hard-pressed decision maker would likely overlook in haste.

Extensive mechanisms have been provided in ARES to encourage manual review and coercion of solutions. However, there have been very few cases in which such guidance improved solutions and many instances in which minor manual adjustments of operational assignments inflicted great disruption. For example, some operational assignments of \((IP_L)\) "cross-locate" units in the sense that a pair of units will each be collocated with a task assigned to the other. This superficial blemish can easily be masked by manual intervention or by automated solution editing. Surprisingly, the removal of cross-locations frequently increases the logistic cost of the solution: there is a very delicate balance of logistic support of task cohorts assigned to specialized units. Cross-location can actually make a great deal of sense in practice.

Manual intervention can work well in cases inviting human judgement. For instance, nearly completed tasks or tasks which have been in progress for long intervals can enjoy efficiencies not apparent to our models. The decision maker can easily declare tasks completed when minor requirements remain, or when it is clear that the models are unduly influenced by a minor requirement.
Operational assignments can be restricted so that units are not moved from their initial new locations until the work in their logistic influence has been completed. Surprisingly, this restriction is rarely needed in practice, and in those cases in which multiple relocations are indicated great efficiencies accrue to the mission as a whole. We view this insight as a strong validation of the modelling philosophy underlying ARES.

Fortuitous design decisions to separate operational assignment and tactical allocation models, to decompose time intervals, and to couple the resulting restricted components with simulation and human intervention options have yielded more than the intended benefits. Our original motives were to capture as much reality as possible while still rendering models capable of quick, responsive solution.

The decomposed design also naturally accommodates features which are otherwise difficult to provide. For instance, partial orderings within tasks can be introduced. Also, discussions with Professor Wayne Hughes have suggested the technical feasibility of campaign analysis, two-sided gaming, and force-on-force applications of ARES. In these contexts, the coupling with simulation enhances our capabilities enormously.

ARES was originally designed for use on an IBM-PC. There is no compelling reason not to use this microcomputer, but we encountered a few practical limitations. An arbitrary 640 kilobyte memory region limitation and crippling errors in the FORTRAN compilers available for the IBM-PC present artificial conversion costs which we are not willing to bear. When these unfortunate shortcomings of the IBM-PC are repaired, conversion might be reconsidered. Calibration tests project solution times on the
order of 2 minutes per time interval on IBM-PC/AT with a math co-processor and internal clock speed of 8 megahertz.
LIST OF REFERENCES


Department of the Army, Field Manual FM 101-10-1: Staff Officers'
Field Manual Organizational, Technical, and Logistical Data,


### INITIAL DISTRIBUTION LIST

<table>
<thead>
<tr>
<th>No.</th>
<th>Copies</th>
<th>Distribution Details</th>
</tr>
</thead>
</table>
| 1.  | 2      | Defense Documentation Center  
Cameron Station  
Alexandria, Virginia 22304-6145 |
| 2.  | 2      | Library, Code 0142  
Naval Postgraduate School  
Monterey, California 93943-5002 |
| 3.  | 1      | Department Chairman, Code 54  
Department of Administrative Science  
Naval Postgraduate School  
Monterey, California 93943 |
| 4.  | 5      | Professor Gerald G. Brown, Code 55BW  
Department of Operations Research  
Naval Postgraduate School  
Monterey, California 93943 |
| 5.  | 1      | Professor Benjamin Roberts, Code 54RO  
Department of Administrative Science  
Naval Postgraduate School  
Monterey, California 93943 |
| 6.  | 1      | Hellenic Armed Forces Headquarters  
National Defense General Staff  
Department of Studies  
Holargos, Athens, GREECE |
| 7.  | 2      | Hellenic Army Headquarters  
Corps of Engineers Department  
Holargos, Athens, GREECE |
| 8.  | 1      | Hellenic Army Headquarters  
Defense Plans and Programs Department  
Holargos, Athens, GREECE |
| 9.  | 1      | Hellenic Army Headquarters  
Department of Studies  
Holargos, Athens, GREECE |
| 10. | 10     | Major Antonios L. Vassiliou  
7th Parodos Nicopoleos 7  
Ioannina, GREECE |