Optimizing Disaster Relief: Real-Time Operational and Tactical Decision Support

Gerald G. Brown
Naval Postgraduate School, Monterey, California 93943

Antonios L. Vassiliou
Hellenic Army Headquarters, Athens, Greece

"Tell me, Muse, of the man of many resources who wandered far and wide,"

Homer (The Iliad)

We introduce a real-time decision support system which uses optimization methods, simulation, and the judgement of the decision maker for *operational assignment* of units to tasks and for *tactical allocation* of units 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. (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 is designed for use in real time, and quick data preparation is aided by the provision from published sources of standard data for many foreseeable tasks; this data can be quickly accessed via visual icons on a computer screen and customized for the actual work at hand.

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; our approach may also be applicable to the operational art and tactics of warfare, and to related multiperiod, large-scale employment of heterogeneous, substitutable resources restricted in availability and demand over time, over geography, and by organizationallimitations.

Naval Research Logistics, Vol. 40, pp. 1-23 (1993)

CCC *0894.069X/93/010001.23*

Although a great deal of work has been done in strategic modeling in many contexts, there is relatively little available modeling 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.

As an example of the kind of problem we are interested in, we develop a scenario following an earthquake. Based on early damage reports some construction battalions are to be mobilized to repair damage to roads, bridges, drinking water systems, hospitals, and so forth. Each battalion is designed to operate independently, but each has different equipment, skilled manpower, initial location, mobility, etc. Some repair efforts may start almost immediately if at least an advance party can be moved to the damage area, and the tempo of work will increase as the battalion relocates to and gets settled in the damage area. However, complete repair may take weeks and the work required depends upon the damage and the stage of repair. Any particular battalion may be well suited for bridge repair, but ill suited for repairing water mains. A battalion might be moved more than once during repairs, but at some disruption cost.

For this scenario, the immediate operational issues are:

- Where should each battalion be sent?
- Each repair should be the responsibility of which battalion?

We want to give each battalion exclusive responsibility for repairing its portion of the damage, and we want to consider how hard and far the battalion movements will be and how well matched the battalions are to the work they are assigned.

The tactical issues are:

- If a battalion allocates its resources well to the repairs for which it is responsible, how fast will work be completed?
- Can substitution be made among resources to get the work done?

Some work must be completed before other work can start, some damage is more urgent to repair than other damage, and if a battalion is bivouacked some distance from the scene work is necessarily slowed by commuting equipment and manpower. If a battalion lacks a heavy grader, substitution of shovels may eventually do the job.

After initial mobilization, operational and tactical decisions are reviewed as more accurate damage assessments become available and the progress of repair efforts is reported. Changes in battalion responsibilities, even additional relocations, may become desirable. The goal is to complete repairs as quickly as possible.

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 [21]) are explicitly motivated by large-scale logistics problems. Karchere and Hoeber [19] 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 [16] reports RAND's first use of man-machine simulation of logistics support activities. Pritsker, Watters, and Wolfe [22] report another RAND effort proposing discrete linear optimization for multiproject scheduling with constraints on multiple, substitutable resources. Chaiken and Larson [8] 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, whom should they serve; and how can they be relocated to substitute for units not available? Kaplan [18] redeploys divisible resources with linear programming. Fitzsimmons [15] states a nonlinear response-time model and uses pattern search to locate units well and allocate workload equitably. Swoveland, Uyeno, Vertinsky, and Vickson [23] 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 [3] formulate strategic force planning models as two-sided games solved with nonlinear programming. Bracken, Falk, and Karr [2] apply multiperiod, two-person zero-sum games formulated to develop strategies for unit sortie allocations. Finally, Kolesar and Walker [20] develop a multistage 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 data describing possible standardized missions. This faithfully follows standard practice in military planning and in disaster planning. The idea is to help the decision maker quickly assemble a data scenario closely resembling the proximate situation from a computer-screen menu of icons representing each of these foreseeable mission types.

We characterize the mission at hand as a set of geographically dispersed *tasks*, each composed of partially ordered subtasks 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. Following standard doctrine, responsibility for each task rests with only one unit at any given time.

ARES is coordinated by a time-interval *decision support simulator* which scales and manipulates scenario data in a fashion transparent to the decision maker and employs a georeference system, a mobility system, a decision-maker simulator, and extensive user interface and user override and control facilities. Two integer linear programs and a linear program complement the prototypic model 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 starts with the determination of tasks and task attributes derived in large part from standard cataloged data for similar tasks. Next, units are identified which might perform the tasks and unit attributes are established. A georeference system is used to generate distance costs and estimate delays in relocating and operating units. The decision maker may *preview* and modify data or *manually preassign* tasks and units as he sees fit.

The *decision support simulator* acts as a coordinating program between the user and various system and data components, enabling the user to quickly specify a decision scenario for some given time horizon, and ensuring that the user can understand the advice rendered, accept all or any part of it, and apply his own judgment.

Operational assignment of tasks to units uses one of two integer programming models ($\mathrm{IP_L}$) or (IP); these two embedded models render good assignments and serve as examples of other tools which may be developed as needed to expand ARES. Each unit will be given exclusive responsibility for each of a set of tasks. 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 analysis.

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 immediate operational and tactical plan, which he can accept, or modify, or reject outright and reconstruct. The decision maker may even use ARES to simulate his decisions and their effects into the far future to forecast eventual outcomes as a consequence of current actions. Regardless of the course of action, ARES is designed to lend quick insight. The decision maker can use his own judgment, especially concerning nonquantified factors, and should gain a better grasp of the overall situation from ARES.

1. DECISION SUPPORT SIMULATOR

The decision support simulator serves primarily to give the user means to quickly create a scenario, to evaluate alternate action plans, and to keep up with developments as time progresses. It also exploits prior planning efforts by use of standardized task data, unit descriptions, a georeference system, a mobility system, and a decision simulator.

The decision support simulator acts as a coordinating program between the user and various model and data components. Its key role is to enable the user to quickly build a decision scenario for some specified time horizon, and to ensure that the user can understand the advice rendered, accept all or any part of it, and apply his own judgment.

To be used, a decision support system must be understood and accepted by the decision maker. In this vein, we follow common practice of decision makers by simplifying our complex problem by temporal and functional decomposition. That is, we concentrate on near-term requirements first, and suggest overall operational assignment of tasks to units before predicting in detail just how each unit will fare with the work it is assigned. This decoupling renders problem fragments that are easier to grasp, and decisions that are easier to evaluate.

Our decisions bear heavily on some view of the situation in the near future. However, some consideration of farther future outcomes is also necessary. The decision support simulator manipulates a scenario script which shows current location and status of units and tasks. For the proximate time interval, a *view* of the resources available to units and needed by tasks is generated; these estimates can consider sequence dependencies among subtasks, prioritization of tasks, logistics of unit movements, learning effects on unit efficiency, and so forth. The details are tedious but the intent is clear:

- Given the situation as we now understand it, what resources can be brought to bear, and what work should be done?

In the following discussion of embedded models, very simple nonlinear functions illustrate our view mechanism without resorting to excessive detail.

A simple georeference system divides the area of operations into contiguous zones. Each zone is small enough that locations within it can be treated as if they are collocated. Arcs connect local pairs of zones to represent feasible direct point-to-point transportation and bear costs for available modes; the arcs here represent road and rail connections and the costs are transit times for categories of units. Damage to a transportation system is quickly expressed by modifying costs for those arcs affected. This scheme is easily adopted by planners, and can be modified to use standard point location codes (SPLCs), or other georeferent keys.

Mobility can be modeled by path finding in the georeference network for modes feasible for the units to be moved. One would expect each unit to suffer some initial delays in marshalling resources, and some subsequent delays in arrival and resettlement in a new location. For construction battalions, heavy equipment is a major impediment. We omit excessive logistic details: The key idea here is to help planners develop a sufficiently detailed model of mobility before the fact, and to be able to quickly modify and use it after the fact.

A decision simulator is provided so that scenarios can be automatically evaluated to an ultimate conclusion. That is, given some scenario warranting such analysis, the decision simulator can perform operational assignment of tasks of units and tactical allocation of unit resources period by period without human intervention. Between decision iterations for each future time period, units are advanced if necessary, task descriptions are modified by the work forecast to be completed, and the scenario script is automatically updated. Interpretation of narrative performance reports for complete decision simulations helps improve local decision rules and lends insight to the overall review and decision process.

The following sections introduce two operational assignment models and then a tactical allocation model. After some discussion, an application of the decision simulator is used to motivate its design.

II. OPERATIONAL ASSIGNMENT MODEL (IP)

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

Index Use

i Tasks
j Resources
k Units

Given Data

d_{ik} Distance cost from unit k to task i
$\underline{r}_{ij}$, $\overline{r}_{ij}$ Minimum, maximum resource j requirements of task i
$\underline{a}_{jk}$, $\overline{a}_{jk}$ Minimum, maximum resource j employable by unit k
$\underline{z}_{jk}$, $\overline{z}_{jk}$ Penalties for violating minimum, maximum resource limits
p_i Priority of task i (>0)
$\underline{u}_i$, $\overline{u}_i$ Penalties for not assigning or double assigning task i
f_{jj} Substitution efficiency of resource j (>0)
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

$$\min \sum_{ik} d_{ik} x_{ik},$$

$$\text{s.t.} \quad \sum_k x_{ik} \overset{\circ}{=} (1, 1); \qquad (\underline{u}_i, \overline{u}_i), \qquad \text{for all } i, \qquad (1) \quad (\text{GUB})$$

$$\sum_i h_{ijk} x_{ik} \overset{\circ}{=} (\underline{a}_{jk}, \overline{a}_{jk}); \qquad (\underline{z}_{jk}, \overline{z}_{jk}), \qquad \text{for all } j, k, \quad (2)$$

$$x_{ik} = \{0, 1\}, \qquad \text{for all } i, k. \quad (3) \quad (\text{IP})$$

The notation $\overset{\circ}{=} (\underline{r}, \overline{r}); (\underline{z}, \overline{z})$ indicates lower and upper ranges $(\underline{r}, \overline{r})$ on row functional values with corresponding respective linear penalties per unit of violation $(\underline{z}, \overline{z})$; i.e., this is a goal program with linear penalties, an elastic integer program (Brown and Graves [5]).

Constraints

(1) Encourage assignment of each task to exactly one unit and form a generalized upper bound (GUB) row set (Dantzig and Van Slyke [9])
(2) express the goodness of fit of task assignments with employable unit resources; this goodness of fit is discussed shortly, and
(3) preclude fractional assignment of tasks of units.

For the proximate time interval, a *view* of the consumption by task i of resource j from unit k is defined:

$$h_{ijk} \equiv \underline{r}_{ij} e^{-\ln f_{ij} + (p_i - 1)/10 + d_{ik}/\sigma_k + t_{ik}^{-2}}, \tag{1}$$

where σ_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 $\underline{r}_{ij}$ to account for the state of resource readiness f_{ij}, the task priority p_i (making less important tasks appear more expensive), the logistic proximity of unit k and task i, d_{ik}/σ_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 judgment of the decision maker. Alternate consumption functions may appeal in other situations.

The distance costs d_{ik} and penalties $\underline{u}_i$, $\overline{u}_i$ and $\underline{z}_{jk}$, $\overline{z}_{jk}$ are expressed in commensurate units and deserve some thought by the modeler. For instance, $\overline{z}_{jk}$ may be interpreted as how much additional distance cost should be incurred before considering overtaxing maximum resource employment $\overline{a}_{jk}$ 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, $\underline{z}_{jk}$ and $\overline{z}_{jk}$ are part of the input script, $\underline{u}_i$ is defined as $100/p_i$, and $\overline{u}_i$ equals 100.

(IP) is intended to quickly assemble aggregate sets of tasks which seem from our current problem view to make good cohorts for particular units. The goodness of fit of such aggregations for particular units can be developed in ways other than ours, and the actual formation of assignments can be carried out by applying alternate models or heuristics. Based on our experience competing this model against unaided decision makers, and reconciling differences between outcomes, (IP) renders advice similar to human decisions.

(IP) is provided as an embedded function within the decision support simulator.

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
$\bar{r}_{jil}$ Gross resource requirement j of task i performed from location l
$\bar{a}_{jlk}$ 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

$$\min \sum_i \sum_l g_{il} z_{il} + \sum_k \sum_l d_{lk} x_{lk},$$

$$\sum_l z_{il} \overset{\circ}{=} (1, 1); (\underline{u}_i, \overline{u}_i), \qquad \text{for all } i, \quad (1) \quad \text{(GUB)}$$

$$\sum_l x_{lk} \overset{\circ}{=} (1, 1); (m, m), \qquad \text{for all } k, \quad (2) \quad \text{(GUB)}$$

$$\sum_k x_{lk} \overset{\circ}{=} (0, 1); (m, m), \qquad \text{for all } l, \quad (3)$$

$$-z_{il} + \sum_k x_{lk} \overset{\circ}{=} (0, 1); (m, m), \qquad \text{for all } i, l, \quad (4)$$

$$-\sum_i \bar{r}_{jil} z_{il} + \sum_k \bar{a}_{jlk} x_{lk} \overset{\circ}{=} (0, 0); (\underline{b}, \overline{b}), \qquad \text{for all } l, j, \quad (5)$$

$$z_{il} = \{0, 1\}, \qquad \text{for all } i, l, \quad (6)$$

$$x_{lk} = \{0, 1\}, \qquad \text{for all } l, k. \quad (7) \quad (\text{IP}_\text{L})$$

(IP_L) uses the notation of (IP). Constraints

(1) encourage assignment of each task to some location,
(2) allow movement of each unit to some location [a GUB row set is formed by constraints (1) and (2)],
(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 (i.e., a good fit), and
(6) and (7) preclude fractional location of tasks and units.

For the proximate time interval, a *view* of the gross resource requirement $\tilde{r}_{jil}$ represents the resource j estimated to be required at location l in order that task i actually receive $\underline{r}_{ij}$:

$$\tilde{r}_{jil} \equiv \underline{r}_{ij} e^{-\ln f_{ij} + (p_i - 1)/10 + g_{il}/\rho}. \tag{2}$$

where ρ expresses the logistic radius of influence from any location; we have used $\rho = 100$. 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:

$$\tilde{a}_{jlk} \equiv \overline{a}_{jk} f_{jj} (\alpha_{lk} + (1 - \alpha_{lk}) e^{-d_{lk}/\sigma_k}), \tag{3}$$

where α_{lk} is the fraction of time which the unit will spend at its destination location and σ_k is the speed of advance.

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

(IP_L) can relocate units, unlike (IP). Each aggregate set of tasks is to be performed by a unit relocated for that purpose. The goodness of fit of such aggregations depends upon task locations, unit locations, and our view of how well such movements can be carried out and assigned work performed.

The decision support simulator provides (IP_L) as an embedded function. In our experience, the decision maker may prefer (IP_L) in early stages of a scenario, and (IP) later.

IV. TACTICAL ALLOCATION MODEL (GN_k)

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

$\underline{r}_{iw}$, $\overline{r}_{iw}$ Minimum, maximum work requirements w of assigned task i
$\underline{q}_{iw}$, $\overline{q}_{iw}$ Penalties for violating minimum, maximum work requirements
$\underline{a}_{jk}$, $\overline{a}_{jk}$ Minimum, maximum resource j employable by unit k
$\underline{z}_{jk}$, $\overline{z}_{jk}$ 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

$$\max_{\text{s.t.}} \sum_i \sum_w \sum_j e_{iwj} y_{iwj},$$

$$\sum_j y_{iwj} \ \hat{=}\ (\underline{r}_{iw}, \overline{r}_{iw}); (\underline{q}_{iw}, \overline{q}_{iw}), \qquad \text{for all } i, w, \qquad (1) \quad (\text{GUB})$$

$$\sum_i \sum_w e_{iwj} y_{iwj} \ \hat{=}\ (\underline{a}_{jk}, \overline{a}_{jk}); (\underline{z}_{jk}, \overline{z}_{jk}), \qquad \text{for all } j, \qquad (2)$$

$$y_{iwj} \geq 0, \qquad \text{for all } i, w, j. \quad (3) \quad (\text{GN}_k)$$

(GN_k) uses the notation of (IP). However, the dimensions of (GN_k) discriminate between resources consumed j and the work completed w, explicitly representing substitutability of resources. Constraints

(1) encourage allocation of sufficient work resources, while
(2) indicate the desired mix of employable unit resources, and
(3) require nonnegative resource allocations.

(GN_k) is an elastic generalized network (traditional generalized networks are discussed by Brown and McBride [7]; the elastic extension is subsequent, unpublished work by Brown).

For the proximate time interval, a *view* of the efficiency of resource j used for work w on task i is defined

$$e_{iwj} \equiv e^{-\ln f_{jw} + (p_i - 1)/10 + s^+_{iw}/10 + \overline{d}_{ik}/\sigma_k}, \tag{4}$$

where $s^+_{iw} = \max\{0, s_{iw} - t\}$, t is the last time period of this allocation, and σ_k is the speed of advance of unit k. The efficiency (4) employs the readiness and substitutability of resources via f_{jw}. 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,

$$\overline{d}_{ik} \equiv \max\{0, d_{ik} - \sigma_k/2\}. \tag{5}$$

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

$$\overline{d}_{ik} \equiv \max\{0, d_{ik} - \sigma_k/2\} + g_{il}. \tag{6}$$

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

The decision support simulator provides (GN_k) as an embedded function.

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.

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

Tactical allocation models (GN) are given unit and task assignments and planned unit movements. Therefore, (GN) 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 [14] and task standards in [11].

For each unit, a speed of advance (SOA) σ_k is given; if a relocation of a unit has been ordered, the new location index (LL) and the time period (PP) of its

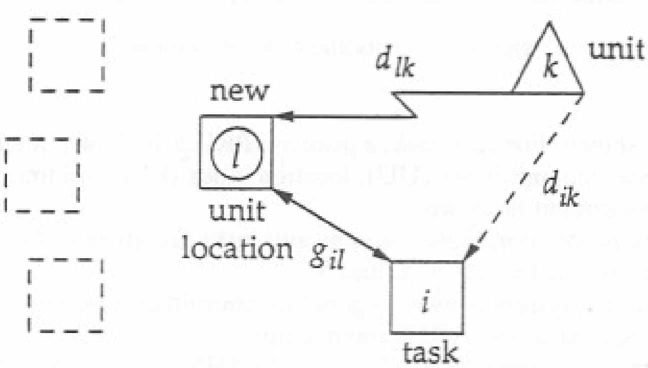

Figure 1. Idealized geographic logistical scenario.

UNIT LABELS, LOCATIONS, AND PRIOR ASSIGNMENTS (IPL)

UU	LABEL		X-COORD	Y-COORD	SOA	LL	PP
1	1ST COMBAT BN	5-35	20.30	20.30	150		
2	2ND COMBAT BN	5-35	04.50	05.50	150		
3	3RD COMBAT BN	5-155	07.00	17.25	150		
4	4TH COMBAT BN	5-155	04.90	08.70	150		
5	1ST CONSTR BN	5-115	05.20	14.50	100		
6	2ND CONSTR BN	5-115	13.30	18.75	100		
7	3RD CONSTR BN	5-115	11.05	15.60	100		
8	4TH CONSTR BN	5-115	08.50	15.25	100		
9	1ST AIRBOR BN	5-195	13.00	05.00	200		
10	2ND AIRBOR BN	5-195	20.30	20.30	200		
11	1ST LIGH.EQUI.CO	5-58	00.40	12.85	150		
12	1ST ENG ARM BT	5-145	13.20	08.60	120		
13	2ND ENG ARM BT	5-145	09.25	02.90	120		
14	3RD ENG ARM BT	5-145	08.50	15.25	120		

TASK LABELS, LOCATIONS, PRIORITIES, AND PRIOR ASSIGNMENTS (IP)(IPL)

TT	LABEL		X-COORD	Y-COORD	PRI	UU	LL	PP
1	ADMIN BUILDING	AA1051	09.55	05.90	1			
2	ADMIN BUILDING	AA1051	09.60	11.60	1			
3	ADMIN BUILDING	AA1101	01.90	06.95	1			
4	HOSPITAL 100 BED	GH0111	10.75	07.45	1			
5	HOSPITAL 200 BED	GH0211	09.60	11.60	1			
6	HOSPITAL 100 BED	GH0131	09.55	05.90	1			
7	HOSPITAL 100 BED	GH0131	01.90	06.95	1			
8	RAILROAD BRIDGE	861643	09.70	05.90	1			
9	RAILROAD BRIDGE	861512	07.90	09.45	1			
10	ROAD BRIDGE 50'	854101	10.70	07.20	2			
11	ROAD BRIDGE 100'	854109	09.70	05.89	1			
12	ROAD BRIDGE 70'	854104	08.30	09.30	2			
13	ROAD 3.5 MILES	853120	10.80	06.15	2			
14	ROAD 4.7 MILES	853122	10.76	07.50	2			
15	ROAD 5.5 MILES	853128	09.60	06.30	2			
16	ROAD 6.8 MILES	853124	02.35	07.25	2			
17	ROAD 6.0 MILES	853120	10.80	06.95	2			
18	WATER TANK-DIST-SUP	NO1	10.85	06.90	1			
19	WATER TANK-DIST-SUP	NO2	09.65	06.35	1			
20	WATER TANK-DIST-SUP	NO3	09.10	10.85	1			

Figure 2. Units and tasks of example.

selection are shown. For each task, a priority (PRI) $\underline{p_i}$ is shown; if an assignment has been made, the unit index (UU), location index (LL), and time period (PP) of the last assignment is shown.

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

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, appear in Figure 6, a segment of the input script. Resource

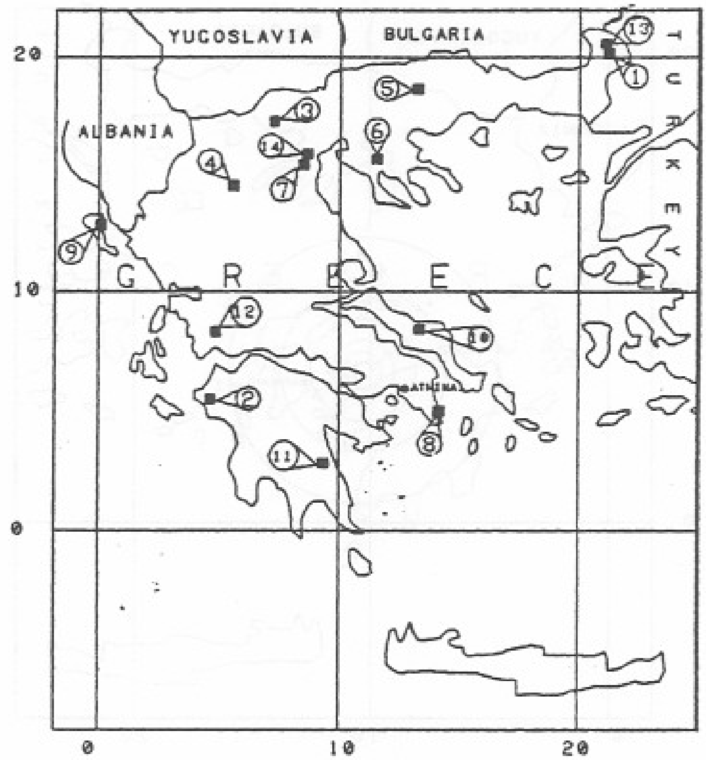

Figure 3. Initial geographic locations of units. (Coordinates displayed are a georeference in common with the following figures.)

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 [11–13]). We envision a taxonomy of standardized task data 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.

The resources employable by Unit 1 ("1ST COMBAT BN"), a combat engineering battalion, are shown in Figure 7, another segment of the input script. These resource endowments are in line with those given by the United States Department of the Army [11] with conversion to man hours from [10]. Penalties for under- or overutilization of resources are also shown.

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.

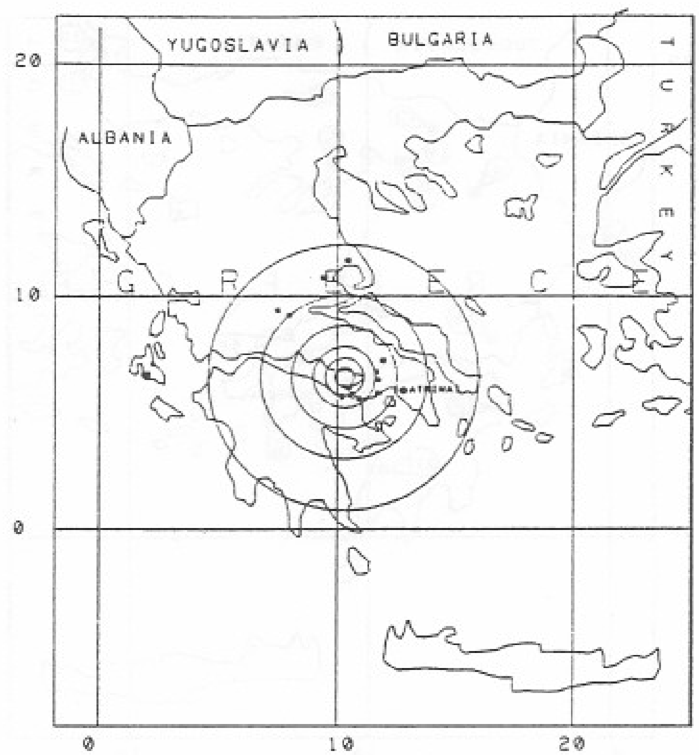

Figure 4. Earthquake epicenter.

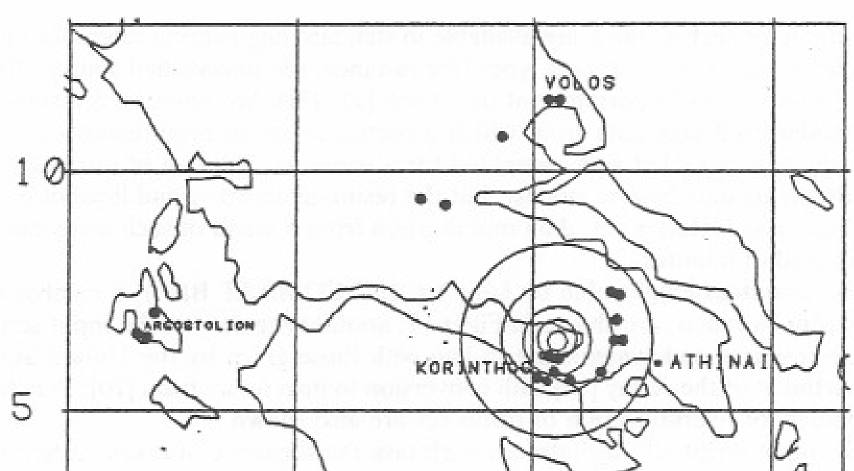

Figure 5. Geographic locations of tasks. (Geographic locations of damaged public works and earthquake epicenter are shown.)

RESOURCE LABELS AND TASK 1 REQUIREMENTS

RR	ILABEL	MAN HOURS
1	ENGIN-PION-APREN-HLPER	6648
2	SURVEYOR	70
3	CARPENTER	7557
4	ELECTRICIAN	940
5	PLUMBER	1740
6	MASON	1600
7	STRUCTURE SPECIAL.	0
8	HEAT-VENTILAT SPECIAL.	200
9	WELDER	0
10	PIPELINE	0
11	CRANE-SHOVEL OPER.	200
12	LOADER OPER.	350
13	DOZER OPER.	300
14	COMPRESSOR OPER.	0
15	DUMP TRUCK OPER.	400
16	CONCRETE MACHINE OPER.	0
17	GRADER OPER.	329
18	CRUSHER OPER.	0
19	DITCH MACHINE OPER.	300
20	ASPHALT SPECIAL.	0
21	POWER ROLLER OPER.	0
22	WATER DISTRIBUT. OPER.	0
23	POWER BOAT OPER.	0
24	ROTARY TILLER OPER.	0
25	SCRAPER OPER.	0

Figure 6. Resource requirements of Task 1.

VII. DESIGN AND IMPLEMENTATION

ARES is intended to help the decision maker, not to replace him. Figures 8 and 9 show 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.

ARES is implemented in FORTRAN and runs on many computers. The results reported here employ IBM VS-FORTRAN 77/2 on an AMDAHL 5995-700 computer using the VM/CMS operating system. (Software copyrights IBM Corporation.) ARES also runs on a microcomputer (e.g., Intel 386 or 486, Microsoft MS/DOS 5.0, using SVS Language System 386 DOS Extended Environment, software copyright Silicon Valley Software). Input scripts may be viewed and edited by a full-screen editor, or imported from spreadsheet or database hosts (e.g., Microsoft's EXCEL). An interactive graphical interface has been designed for ARES using XVT (copyright XVT Corporation); however, based on our experience with similar applications, several more months would be required to implement this system and custom tailor it to suit a given user of a particular host computer.

ARES uses the X-SYSTEM (Brown and Graves [5]) to solve (IP_L), (IP), and (GN_k) in real time. For each problem instance, problem generators directly

RESOURCE LABELS AND UNIT 1 AVAILABILITIES

RR	LABEL	I	MIN	MAX	MIN PEN	MAX PEN
2	SURVEYOR		405	450	10	10
3	CARPENTER		1215	1350	10	10
4	ELECTRICIAN		203	225	10	10
5	PLUMBER		1418	1575	10	10
6	MASON		1215	1350	10	10
7	STRUCTURE SPECIAL.		0	0	10	10
8	HEAT-VENTILAT SPECIAL.		0	0	10	10
9	WELDER		187	208	10	10
10	PIPELINE		0	0	10	10
11	CRANE-SHOVEL OPER.		1215	1350	10	10
12	LOADER OPER.		6165	6850	10	10
13	DOZER OPER.		4050	4500	10	10
14	COMPRESSOR OPER.		1013	1125	10	10
15	DUMP TRUCK OPER.		10935	12150	10	10
16	CONCRETE MACHINE OPER.		203	225	10	10
17	GRADER OPER.		1620	1800	10	10
18	CRUSHER OPER.		0	0	10	10
19	DITCH MACHINE OPER.		0	0	10	10
20	ASPHALT SPECIAL.		0	0	10	10
21	POWER ROLLER OPER.		0	0	10	10
22	WATER DISTRIBUT. OPER.		0	0	10	10
23	POWER BOAT OPER.		0	0	10	10
24	ROTARY TILLER OPER.		0	0	10	10
25	SCRAPER OPER.		0	0	10	10

Figure 7. Resource Endowment of Unit 1.

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 the decision simulator which performs "judgment 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 decision simulator 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

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 (GN_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 GO TO NEXT_PERIOD

Figure 8. ARES functional specification of decision support simulator.

[4]). 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 modeling arena.

VIII. SCENARIO RESULTS

ARES has been used in simulation mode to completely plan mission scenarios from start to finish. For the earthquake scenario, Figure 10 shows the initial operational assignments of (IP_L). Figure 11 depicts the arrival of units to their initially assigned locations.

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

Face validity of the decision simulator solution has been judged by reviewers who are experienced Army engineers. Manual intervention does not seem to improve solution quality significantly. 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 12 for the seven single-period time intervals to complete the earthquake scenario.

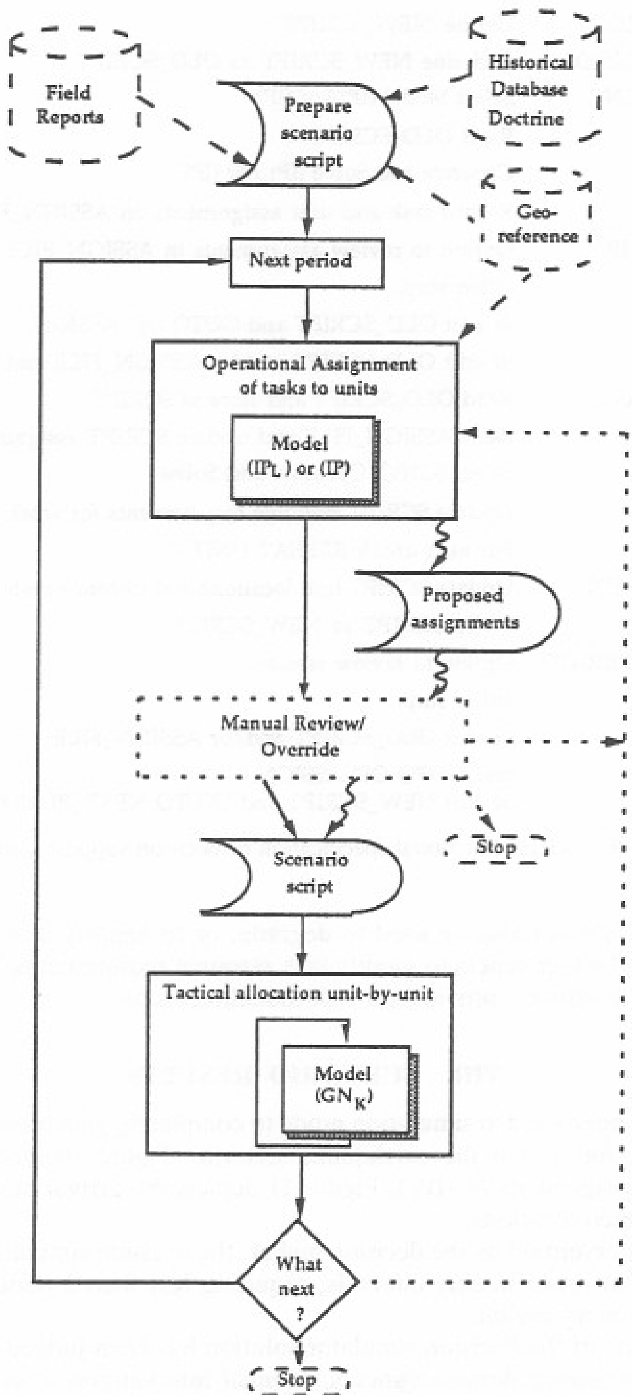

Figure 9. ARES design-overview. Latest situation reports are received and organized in a scenario script, which may also draw from historical data or doctrine. Optimal operational assignment of tasks to units are fused into an updated script, and tactical allocation estimates the current period progress unit by unit. This process may be allowed to continue forecasting into future periods, with or without optional manual intervention depicted by dotted features.

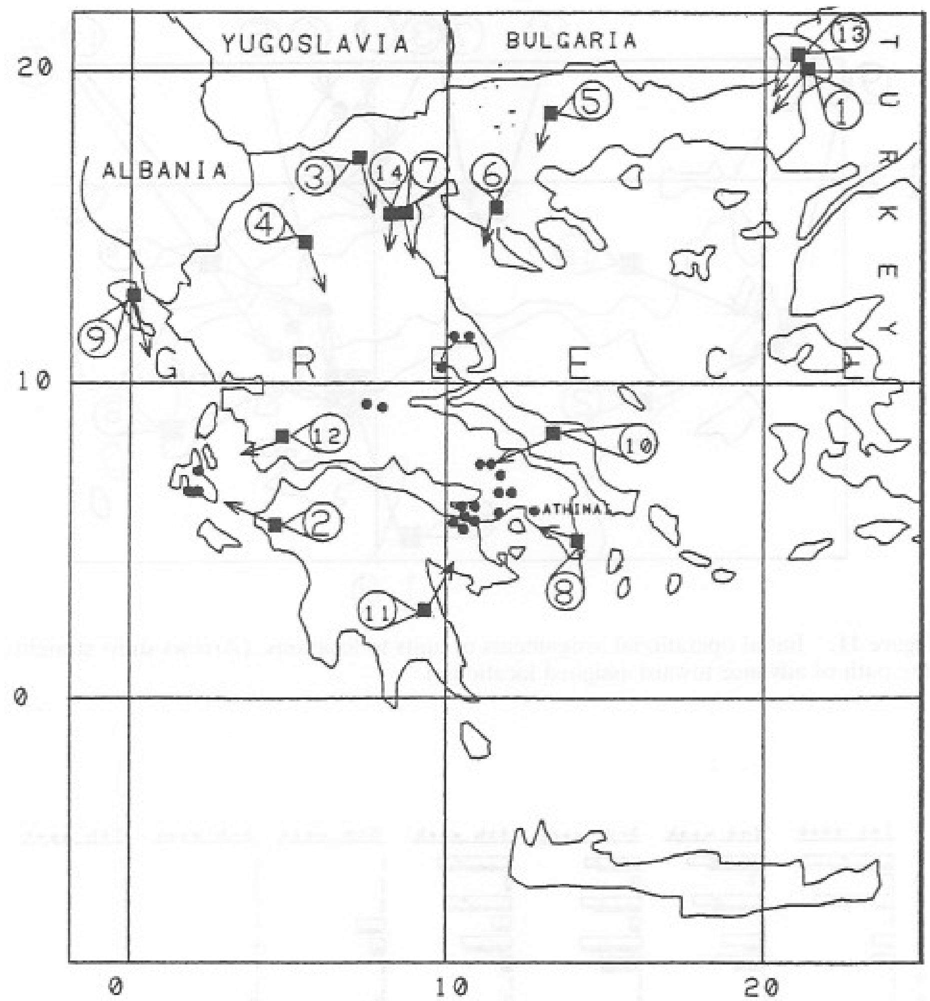

Figure 10. Initial operational assignments of units. (Directional vectors show the straight-line path and relative speed of advance σ_k.

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 nonzero density of resource availabilities and remaining requirements. (IP) typically has about 340 constraints, 268 binary variables, and 6200 nonzero consumption coefficients. The linear program continuous relaxation can be generated and solved in about a second, and an optimal binary solution is achieved in another second, or so.

(IP_L) has about 1000 constraints, 645 binary variables, and 8000 rather unwieldy nonzero gross resource requirement and net resource availability coefficients.

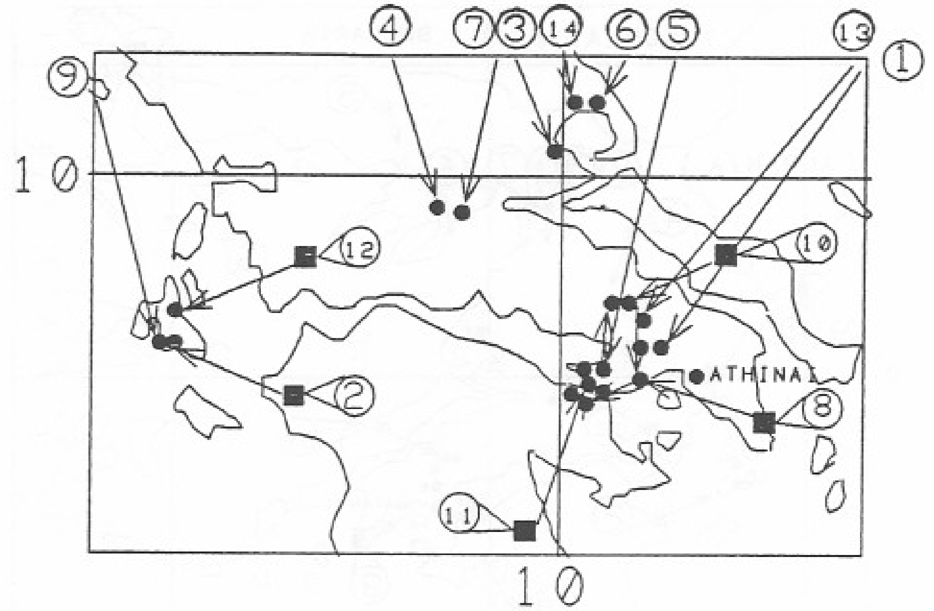

Figure 11. Initial operational assignments of units to locations. (Arrows show straight-line path of advance toward assigned locations.)

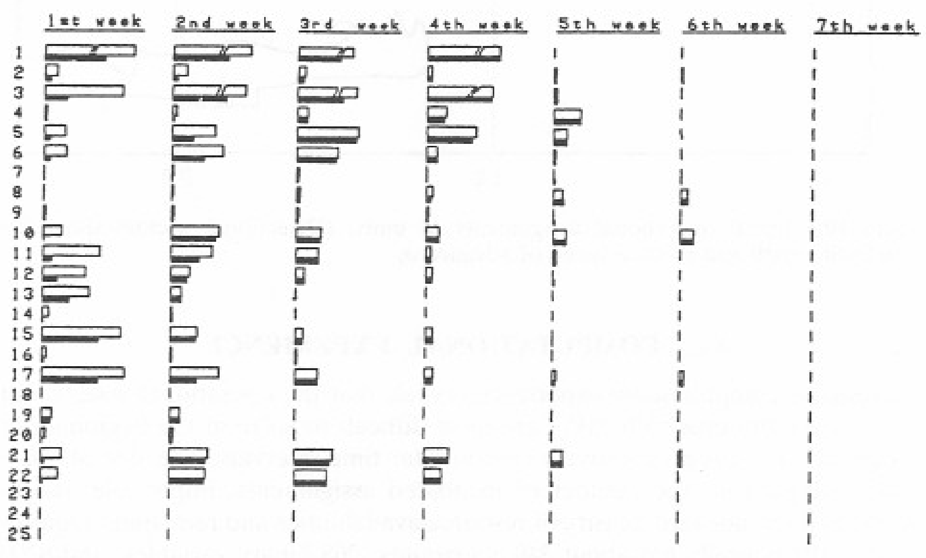

Figure 12. Resource requirements and work completed. (Each row represents a resource requirement over time-interval columns. The white bars depict resource requirements by time interval; the black bars show the relative fulfillment of the requirements. Broken bars are out of scale. From each time interval to the next the requirements are reduced by the work completed and amplified by new sequence-dependent requirements. In this scenario, seven weekly time intervals are required to complete all tasks.)

The linear program continuous relaxation of (IP_L) proved impossible to solve by direct assult. Prior work by Brown and Graves for Bausch [1] on large-scale set-partitioning problems and later refinements by Brown, Graves and Ronen [6] suggested an alternate means of attack: a problem cascade.

Briefly, the rows of constraints and columns of variables are lexicographically sorted to place short rows first accompanied by other rows and columns with intersection nonzero 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.

(IP_L) resisted even the problem cascade until a new heuristic cascade strategy was adopted which activates the shortest $\frac{1}{2}$ of constraints and their associated variables, then the shortest $\frac{3}{4}$, then $\frac{7}{8}$, 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 2 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 their mission 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 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 nonquantifiable guidance in this fashion, but cannot 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 (IP_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. (IP_L) is worth the computational investment.

By contrast, the tactical allocation models (GN) are easy to solve even in the cases where heroic substitution of resources are required. The size of each (GN_k) varies with the number of tasks assigned to the unit, and the nonzero densities of resource availabilities, remaining requirements, and allowable substitutions. For our scenario, a typical instance of (GN_k) has about 70 constraints and 1190 variables, and is generated and solved in less than 0.04 second. Stress tests with 525 constraints and 12,500 variables require less than a second.

X. 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 appear 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 experimentation with 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 judgment. 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 modeling 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 decision 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 (e.g., Hughes [17]). In these contexts, the coupling with simulation enhances our capabilities enormously.

REFERENCES

[1] Bausch, D.O., "Computational Advances in the Solution of Large-Scale Set Covering, and Set Partitioning Problems," M.S. Thesis, Naval Postgraduate School, 1982.

[2] Bracken, J., Falk, J.E., and Karr, A.F., "Two Models for Optimal Allocation of Aircraft Sorties," *Operations Research*, **23**, 979–995 (1975).

[3] Bracken, J., and McGill, J.T., "Defense Applications of Mathematical Programs with Optimization Problems in the Constraints," *Operations Research*, **22**, 1086–1101 (1974).

[4] Brown, G.G., Ellis, C.E., Graves, G.W., and Ronen, D., "Real-Time Wide Area Dispatch of Mobil's Petroleum Tank Trucks," *Interfaces*, **17**, 107–120 (1987).

[5] Brown, G.G., and Graves, G.W., "Design and Implementation of a Large-Scale (Mixed Integer, Nonlinear) Optimization System," paper presented at ORSA/TIMS, Las Vegas, November 1975.

[6] Brown, G.G., Graves, G.W., and Ronen, D., "Scheduling Ocean Transportation of Crude Oil," *Management Science*, **33**, 335–346, (1987).

[7] Brown, G.G., and McBride, R.M., "Solving Generalized Networks," *Management Science*, **30**, 1497–1523 (1984).

[8] Chaiken, J.M., and Larson, R.C., "Methods for Allocating Urban Emergency Units: A Survey," *Management Science*, **19**, P110–P130 (1972).

[9] Dantzig, G.B., and Van Slyke, R.M., "Generalized Upper Bounding Techniques," *Journal of Computing and System Science*, **1** 213–226 (1967).

[10] Department of the Army, *Field Manual FM 5-35: Engineers' Reference and Logistic Data*, Washington, DC, April 1971.

[11] Department of the Army, *Technical Manual TM 5-301-1: Army Facilities Components System—Planning (Temperate)*, Washington, DC, September 1973.

[12] Department of the Army, *Technical Manual TM 5-302: Army Facilities Components System Designs*, Washington, DC, September 1973.

[13] Department of the Army, *Technical Manual TM 5-303: Army Facilities, Components, System-Logistic-Data and Bills of Materials*, Washington, DC, September 1973.

[14] Department of the Army, *Field Manual FM 101-10-1: Staff Officers' Field Manual Organizational, Technical, and Logistical Data*, Washington, DC, July 1976.

[15] Fitzsimmons, J.A., "A Methodology for Emergency Ambulance Deployment," *Management Science*, **19**, 627–636 (1973).

[16] Geisler, M.A., "The Simulation of a Large-Scale Military Activity," *Management Science*, **5**, 359–368 (1959).

[17] Hughes, W., *Fleet Tactics*, Naval Institute Press, Annapolis, 1986.

[18] Kaplan, S., "Readiness and the Optimal Redeployment of Resources," *Naval Research Logistics Quarterly*, **20**, 359–368 (1973).

[19] Karchere, A., and Hoeber, F.P., "Combat Problems, Weapon Systems, and the Theory of Allocation," *Operations Research*, **1**, 286–302 (1953).

[20] Kolesar, P., and Walker, W.E., "An Algorithm for the Dynamic Relocation of Fire Companies," *Operations Research*, **22**, 249–274 (1974).

[21] Koopmans, T.C. (Ed.), *Activity Analysis of Production and Allocation*, Wiley, New York, 1951.

[22] Pritsker, A., Watters, L., and Wolfe, P., "Multiproject Scheduling with Limited Resources: A Zero-One Programming Approach," *Management Science*, **16**, 93–108 (1969).

[23] Swoveland, C., Uyeno, D., Vertinsky, I., and Vickson, R., "Ambulance Location: A Probabilistic Enumeration Approach," *Management Science* **20**, 686–698 (1973).

Manuscript received January 4, 1990
Revised manuscript received April 7, 1992
Accepted May 6, 1992