A MODELING/SOLUTION APPROACH FOR OPTIMAL DEPLOYMENT OF A WEAPONS ARSENAL*

Fred GLOVER[1], Darwin KLINGMAN[2] and Nancy V. PHILLIPS[3]

[1] *U.S. West Chair in Systems Science, School of Business, University of Colorado, Boulder, Colorado 80309-0419, USA*

[2] *Hugh Roy Cullen Centennial Chairholder in Business Administration; Director of the Center for Business Decision Analysis; and Professor of Management Science and Information Systems, Graduate School of Business and Computer Sciences, College of Natural Sciences, The University of Texas at Austin, Austin, Texas 78712, USA*

[3] *Assistant Director of the Center for Business Decision Analysis; and Associate Professor of Management Science and Information Systems, Graduate School of Business, The University of Texas at Austin, Austin, Texas 78712, USA*

Abstract

This paper reports a real-world application of a large-scale assignment/allocation mixed-integer program for optimal deployment and targeting of missiles for the U.S. Strategic Air Command. We provide a NETFORM model that reduces the number of zero-one variables of a standard integer programming formulation by more than two orders of magnitude (by factors approaching 500) and a tailored NETFORM software system that solves problems involving 2,400 zero-one variables and 984,000 continuous variables to within 99.9% of optimality in less than one minute on an IBM 4381.

1. Introduction

This article addresses the problem of devising an optimal strategic strike force plan involving an MIRV weapons delivery system for the U.S. Strategic Air Command. While the proposed model/solution approach is specifically for this particular type of missile system, it is applicable to a wide variety of weapon system deployment problems. In general, many weapon system deployment studies involve the assignment

*This research was supported in part by the Center for Business Decision Analysis, the Hugh Roy Cullen Centennial Chair in Business Administration, and the Office of Naval Research under Contract N00014-87-K-0190. Reproduction in whole or in part is permitted for any purpose of the U.S. Government.

of weapons to potential targets in order to develop a strategic strike force plan with the goal of maximizing the value of targets reached (see, for example, [8]). The weapon assignments must satisfy external constraints imposed on the solution, including the accessibility of each target installation to the weapon assigned to it, multiple hit requirements, and desired weapon mixes for different elements of the target system. A typical problem of this kind involves a few thousand target installations and a few hundred force attack elements, each containing a fixed number of weapons.

Within this general formulation, the problem can be subdivided into two subclasses. The first subclass contains those problems in which the accessibility of targets to weapons is highly restricted, reducing feasible options to a point that allows a heuristic algorithm to sift through available alternatives and provide a satisfactory solution. (This work is detailed in [1].) For the second subclass of problems, target asscessibility is not so severely constrained, giving rise to a decision problem that is highly combinatorial in nature. In this case, the direct heuristic approach suffers major shortcomings, and a more global procedure is required. This paper reports the design of a NETFORM (network-related formulation) and an associated solution procedure tailored for the resulting model structure. The NETFORM approach (including both model and solution method) has proved highly effective, reducing the number of zero-one variables of a standard integer programming formulation by more than two orders of magnitude, and yielding solutions within a fraction of a percent of a global optimality bound, where this fraction becomes progressively smaller as the problem size grows. The method is also highly efficient, requiring less than one minute of CPU time for an IBM 4381 to solve a problem involving 984,000 continuous variables and 2,400 zero-one variables.

2. Initial mixed-integer formulation

The process of weapons delivery may be thought of as delivering a collection of weapon packets, or bundles, to a selected subset of regions called *focus centers*. Each focus center is itself a target, and if a weapon bundle is delivered to a particular center, then one of the weapons in the bundle must be assigned to the focus center as its target. A specified set of other targets is also accessible from the focus center, and the remaining weapons in the bundle can be assigned to any subset of these accessible targets, subject to the provision that at most one weapon is assigned to any given target.

The number of targets accessible from a focus center always exceeds the number of weapons contained in a bundle. At the same time, the targets accessible to two different focus centers may overlap, and a focus center that is sufficiently close to another may indeed be among the accessible targets for the second center. The objective is to choose a collection of focus centers, and for each of these an associated subset of accessible targets, for assigning the weapons of the weapon bundles. More specifically, each target has a rating, depending in part on the focus center from which the target is reached, and the goal is to identify a weapons assignment that maximizes the sum of ratings of the targets reached.

To formulate the force assignment problem mathematically, let X_{ij} and Y_{jk}, respectively, denote the number of weapon bundles (0 or 1) sent from bundle origin i to focus center j, and the number of weapons (0 or 1) sent from focus center j to target k. Let N_1, N_2, and N_3 be the index sets, respectively, for the weapon bundle origins, focus centers, and targets, where the cardinality of N_i is denoted by n_i. Without loss of generality, we assume $n_2 = n_3$ (this is customarily the case but if it were not, we could create dummy focus centers or dummy targets with high cost connections that would preclude their use). We also assume the elements of N_2 and N_3 are indexed so that for each $j \in N_2$, the node j in N_3 with the same index is its associated focus center target. Further, let A_i (B_j) be the set of focus centers (target nodes) accessible from weapon bundle origin i (focus center j), and let C_j (D_k) be the set of weapon bundle origins (focus centers) which have access to focus center j (target k). (Because bundles begin their journey from different locations, not every focus center can be reached by a given bundle origin i.) A suitable cost C_{jk} is defined for each variable Y_{jk} and reflects the rated value of striking target k from focus center j. Finally, T is defined an an integer "target number" measured as a quantity of weapons to be delivered from a weapon bundle origin to a focus center, if weapons are delivered. For our application, T is a constant.

The mathematical formulation is shown below. Constraint (1) ensures that each bundle of T weapons is assigned to exactly one focus center. Constraint (2) ensures that all weapons assigned to a focus center are assigned to targets. Constraint (3) ensures that if a weapon bundle is delivered to a particular focus center, then one of the weapons in the bundle must be assigned to the focus center as its target. Constraints (4) and (5) model the requirements that at most one weapon bundle is assigned to each focus center and at most one weapon is assigned to each target, respectively. It may be observed that inequality (4) is implied by the conjunction of (2) and (3). We have included it, however, to show its role in parallel with the role of (5).

$$\text{Maximize} \quad \sum_{j \in N_2} \sum_{k \in B_j} C_{jk} Y_{jk}$$

$$\text{subject to} \quad \sum_{j \in A_i} X_{ij} = 1, \qquad i \in N_1 \tag{1}$$

$$T \sum_{i \in C_j} X_{ij} = \sum_{k \in B_j} Y_{jk}, \qquad j \in N_2 \tag{2}$$

$$\sum_{k \in B_j} Y_{jk} \leqslant Y_{jj} T , \qquad j \in N_2 \tag{3}$$

$$\sum_{i \in C_j} X_{ij} \leqslant 1 , \qquad j \in N_2 \tag{4}$$

$$\sum_{j \in D_k} Y_{jk} \leqslant 1 , \qquad k \in N_3 \tag{5}$$

$$X_{ij}, Y_{jk} = 0 \text{ or } 1 , \qquad i \in N_1,\ j \in N_2,\ k \in N_3 . \tag{6}$$

Disregarding constraints (3) and (4) leads to the mixed-integer generalized network formulation shown in fig. 1 and referred to as the force assignment problem. The left-most column of nodes (Level 1) represents weapon bundle origins, the center nodes (Level 2) represent focus centers, and the right-most nodes (Level 3) represent targets. Each weapon bundle origin node has a supply of one weapon bundle, shown in the triangle pointing toward the node. Each origin node i is connected by an arc to each focus center node j which is an accessible focus center for origin i. Associated with each of these arcs is the "target number" T, shown in a triangle, which denotes the arc multiplier in generalized network terminology [2,4]. This multiplier represents the fact that T weapons reach focus center node j by each arc that carries a weapon bundle to that node. Each focus center node j is connected by arcs to all target nodes k that are accessible to focus center j. Arc costs (strike ratings) C_{jk} are shown in rectangles on these arcs. Each target node has a demand of at most one weapon, indicated by the arc with 0, 1 bounds in parentheses leaving each target node.

Each arc that is constrained to carry a discrete (integer) flow of 0 or 1 is indicated by appending an asterisk to its bounds, i.e. by the symbol $(0, 1)^*$. Although eq. (6) indicates that all variables are required to be integer (0 or 1), this restriction is partially redundant by reference to the NETFORM of fig. 1. More precisely, this NETFORM reveals hidden unimodularity in the mathematical formulation (1)–(6). To see this, first note that in any problem that consists of a generalized network together with discrete flow conditions, where all problem data other than costs (i.e. bounds, multipliers, supplies, demands) are integers, it is only necessary that arcs with non-unit multipliers must have integer flows to ensure that all arcs will have integer flows. This follows from the fact that by fixing the discrete flows at integer values, the residual problem will satisfy the unimodular extreme point property, and hence any procedure that finds an extreme point solution for the continuous portion of the problem will obtain an all-integer solution. We will now demonstrate that this unimodular property continues to hold in the more complex situation where the side constraints not included in the diagram are incorporated into the model.

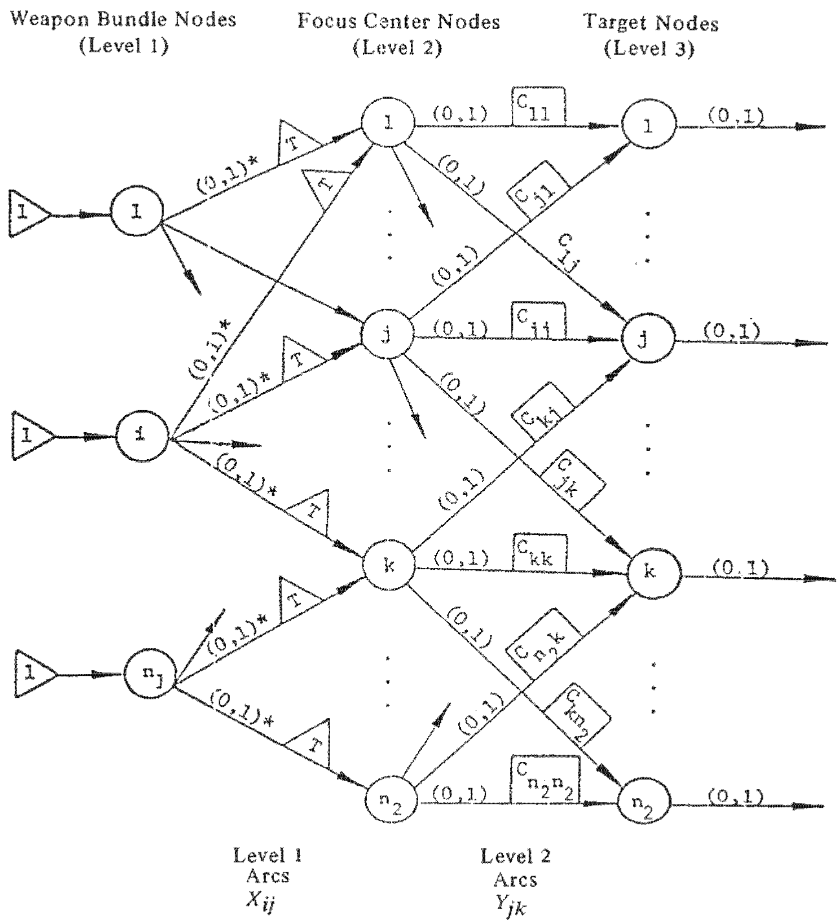

Fig. 1. Mixed-integer generalized network for the force assignment (excluding constraints (3) and (4)).

Within the context of the mixed-integer generalized network formulation, the interpretation of the side (non-network) constraint (3) is that each *horizontal arc* linking a focus center node j to its "companion" target node with the same index is required to carry flow of one unit if any flow enters the focus center node. Furthermore, constraint (3) in conjunction with constraint (2) (where the latter is embodied within the network) ensures that when $Y_{jj} \leqslant 1$ and all X_{ij} are restricted to zero or one, at most one arc entering the focus center node will carry a positive flow, and this flow must deliver exactly T units to the node and force $Y_{jj} = 1$. Thus, when the network in fig. 1 is solved together with these side constraints, and the arcs indicated in the diagram by asterisks are restricted to carry integer flows, all other arcs will carry integer flows automatically. The chief consequence of the insights provided by the NETFORM is the ability to dramatically reduce the number of zero-one variables. Given the integer restrictions on the X_{ij} variables, all other variables can be treated as continuous variables.

3. An improved NETFORM model

Further analysis with the aid of a network-related representation yields a new NETFORM, which in turn provides an improved mathematical formulation. By means of this model, it is possible to achieve a further significant reduction in the number of discrete (0 or 1) variables, while at the same time incorporating all constraints of the previous model into the NETFORM. Both of these types of changes are important for developing an effective solution procedure.

As a basis for the alternative representation, we again make use of the fact that if an arc in fig. 1 carries a positive flow into focus center node j, then by the restrictions (2) and (3), only one arc can do so, and exactly T units of flow must enter node j. Moreover, one of these T units of flow must go on the (j, j) arc from

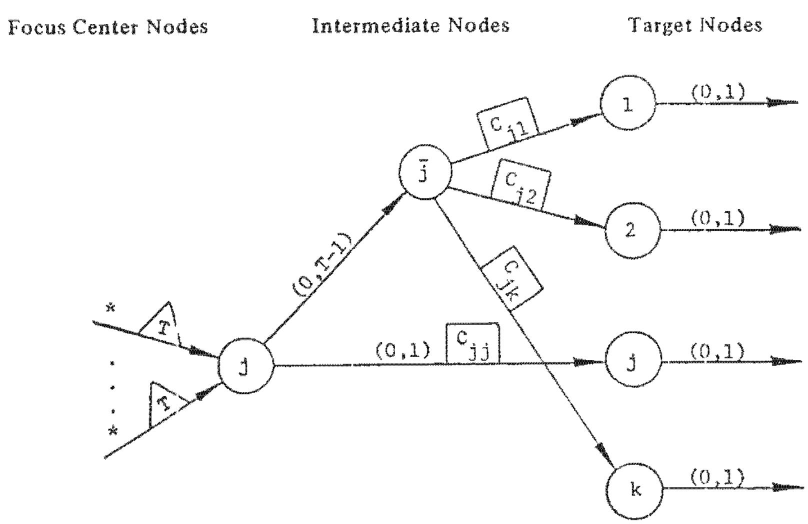

Fig. 2. Network component incorporating companion target requirements and one bundle per focus center into force assignment network.

the focus center node to its associated target, which means that at most $T - 1$ units of flow can go on all other arcs out of focus center node j. The implications of these observations can be clarified and expressed in a straightforward fashion by the diagram in fig. 2, in which the arcs out of focus center node j that lead to targets other than target j are reached by first traversing an intermediate arc to a new node $\bar{j}$.

The indicated modification successfully absorbs conditions that were initially external to the original NETFORM into the network structure. Pictorially, however, we can observe an additional interesting feature that makes it possible to do better. Since only one of the arcs into focus center node j will be able to deliver its load of T units, note that we may intercept the flows into node j by inserting an arc prior to this node that will receive the flows from all arcs that previously entered node j. To embody the desired restrictions, this intermediate arc must be bounded to limit the incoming flow to be at most one unit and must be given a multiplier to translate a unit flow into a load of T units. Letting the initial node of this interposed arc be denoted by $\hat{j}$, we obtain the diagram in fig. 3.

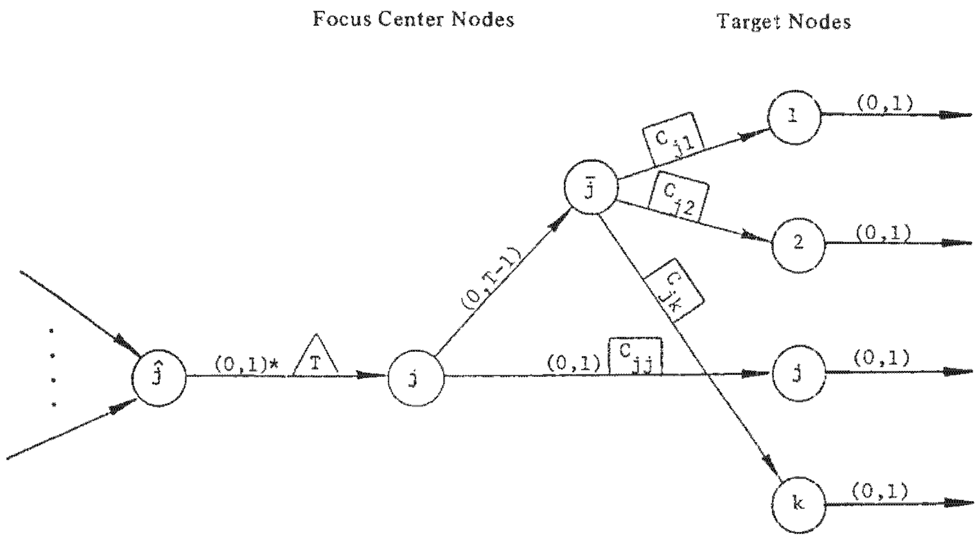

Fig. 3. Network component reducing the number of discrete variables in the force assignment network to one per weapon bundle.

The arcs into node $\hat{j}$ are the same as those previously into node j, except that their multipliers, bounds, and integrality restrictions are removed. The new arc from $\hat{j}$ to j provides the multiplier effect and is restricted to be integer-valued, hence in this case 0 or 1. The power of the changes leading to fig. 3 is that the integer requirement for this single new type of arc from $\hat{j}$ to j can replace the full collection of integer requirements for the arcs in fig. 1. The accumulation of model refinements provided by these visual analyses is shown in fig. 4.

The NETFORM of fig. 4 leads naturally to a new mathematical formulation in which (1) and (5) are retained, and the constraints (2), (3), (4), and (6) are replaced by (7), (8), (9), (10), (11), and (12), as follows. In this representation, the set $C_{\hat{j}}$ equals the previous C_j and the set $B_{\bar{j}}$ equals the previous B_j.

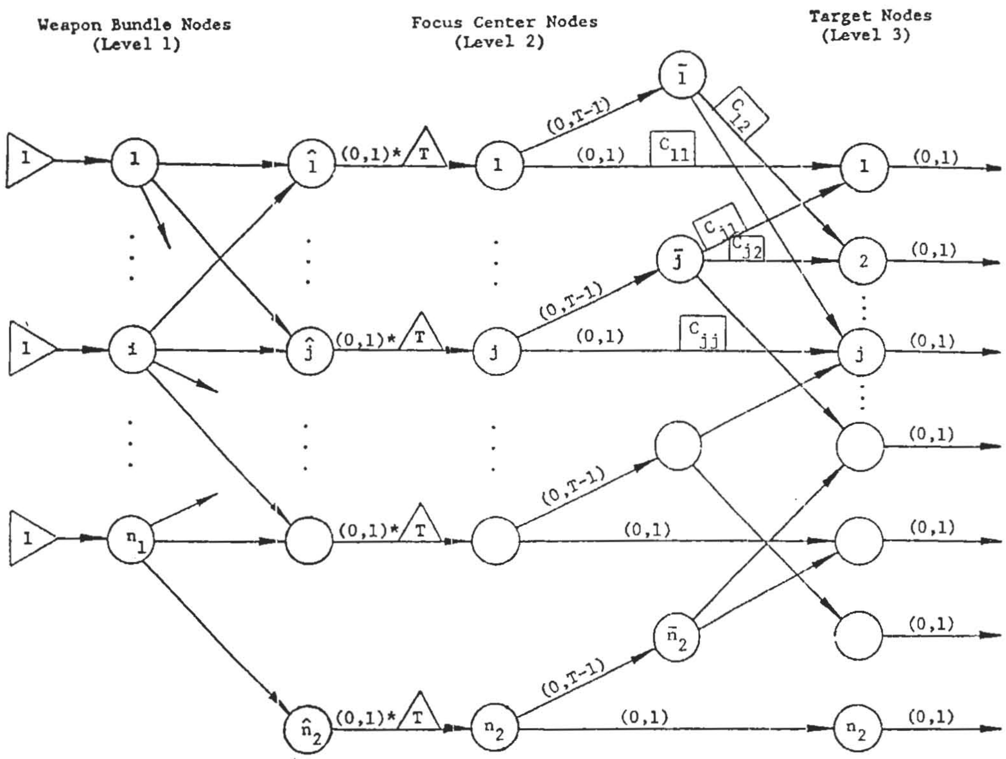

Fig. 4. Mixed-integer generalized force assignment network incorporating all side constraints and reducing number of discrete variables.

$$X_{\hat{j}j} = \sum_{i \in C_{\hat{j}}} X_{i\hat{j}} \,, \qquad \hat{j} \text{ the companion of } j \in N_2 \tag{7}$$

$$T X_{\hat{j}j} = Y_{j\bar{j}} + Y_{jj} \,, \qquad j \in N_2 \tag{8}$$

$$Y_{j\bar{j}} = \sum_{k \in B_{\bar{j}}} Y_{\bar{j}k} \,, \qquad \bar{j} \text{ the companion of } j \in N_2 \tag{9}$$

$$0 \leqslant Y_{jj} \leqslant 1 \,, \qquad j \in N_2 \tag{10}$$

$$0 \leqslant Y_{j\bar{j}} \leqslant T - 1 \,, \qquad j \in N_2 \tag{11}$$

$$X_{\hat{j}j} = 0 \text{ or } 1 \,, \qquad \hat{j} \text{ the companion of } j \in N_2 \,. \tag{12}$$

The transformation of the variables provided by this new formulation, which is responsible for the additional gains in reducing the number of integer variables, is not evident from a mathematical standpoint and underscores the importance of the pictorial aid provided by the NETFORM as a means of discovering such relationships.

As noted, the diagram of fig. 4 succeeds in embodying all constraints except for the zero-one integer requirements in a generalized network structure. It is possible to go one step further at the expense of abandoning this property. The trade-off in taking this additional step is to reduce slightly the number of continuous variables and to achieve a slightly stronger problem relaxation when the integer restrictions are disregarded.

The basis for this step is to observe that when an integer-constrained arc from node $\hat{j}$ to node j carries a flow of 1, all arcs meeting node j *simultaneously* carry fully determined flows. Thus, we may replace node j by an AND node, represented by a square rather than a circle, which by convention has the property that all arcs meeting this node carry the same flow (see, e.g. [3]). The appropriate construction is illustrated in fig. 5.

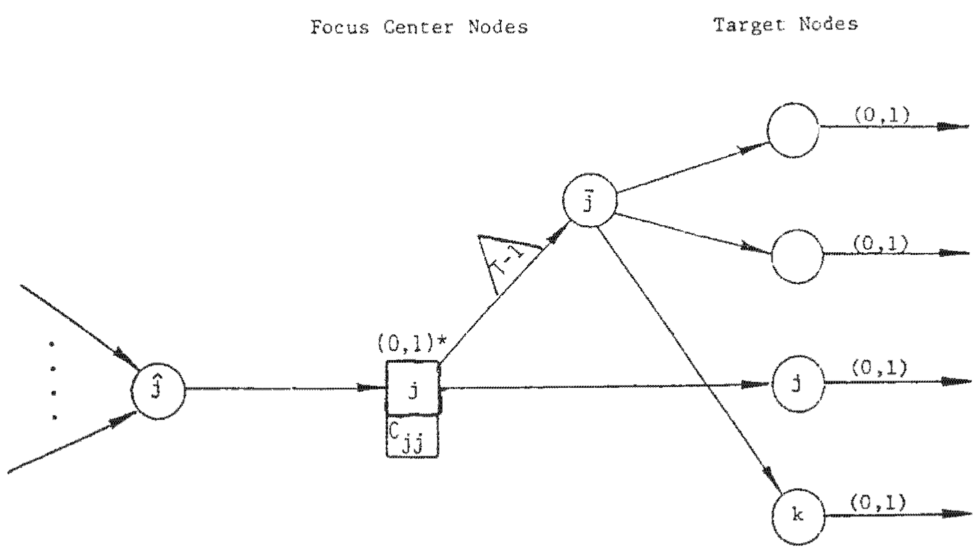

Fig. 5. NETFORM component replacing standard (OR) node of fig. 3 with AND node construction.

In this diagram, the standard node j of fig. 3 (which constitutes an OR node because arcs meeting the node carry flows independently, i.e. disjunctively) is replaced by the corresponding AND node. An AND node and all arcs meeting this node represent a single variable. The constraints directly affected by this AND variable are represented by the standard (circular) nodes which are the peripheral endpoints of its associated arcs. (By the usual convention, each standard node corresponds to a constraint in equality form, stipulating that the flow into the node equals the flow out, allowing for constant terms expressed as supplies and demands.) Accordingly, the three separate arcs (variables) meeting the standard node j in fig. 3 become fused into a single variable in fig. 5. Information relevant to this variable (such as bounds and cost) is now positioned in proximity to the square, which is the embodiment of this variable in the diagram.

The diagram of fig. 5 has the interpretation that each unit of the variable associated with the square simultaneously extracts one unit from node $\hat{j}$, sends $T - 1$ units to node $\bar{j}$, and sends one unit to target node j. This interpretation applies to fractional units as well as to integer units, which yields a stronger problem relaxation than the NETFORM based on fig. 3 when the integer restrictions are dropped. In other respects, the NETFORM based on fig. 5 has the same "structure" as that of fig. 3 and hence we do not bother to display the resulting full problem NETFORM, as the analog to fig. 4, in a separate diagram.

The mathematical formulation corresponding to the NETFORM based on fig. 5 collapses the variables $X_{\hat{j}j}$, $Y_{j\bar{j}}$, and Y_{jj} into a single variable Z_{jj}, and replaces the constraints (7)–(12) by the constraints:

$$Z_{jj} = \sum_{i \in C_j} X_{ij} , \qquad j \in N_2 \tag{13}$$

$$(T - 1)Z_{jj} = \sum_{k \in B_j} Y_{jk} , \qquad j \in N_2 \tag{14}$$

$$Z_{jj} = 0 \text{ or } 1 , \qquad j \in N_2 . \tag{15}$$

Thus, the NETFORM based on using the AND node construction of fig. 5 succeeds in eliminating $3n_2$ constraints and $2n_2$ continuous variables from the formulation based on fig. 3 (or more generally, fig. 4). It should be noted, however, that the consequences of this change are not as widespread as those leading to fig. 4. The number of integer variables is not altered and the number of discarded continuous variables represents a relatively small fraction of the total number of variables. Further, two-thirds of the eliminated constraints are upper bound constraints for problem variables, which do not affect problem sizes for methods that treat such bounds implicitly. Finally, as a comparison of figs. 3 and 5 shows, the connectivity structure of the underlying NETFORMs remains unchanged.

The comparative merits of the NETFORMs based on fig. 3 and on fig. 5 depend on the relative strengths of their continuous relaxations and on their relative exploitability. (Relaxing the integer restrictions on fig. 3 yields a generalized network, while relaxing the integer restrictions on fig. 5 yields a linear programming problem because of the AND node construct.)

Our resolution of this issue in the present context is based on an important attribute of NETFORM representations that remains to be discussed – the capacity to disclose relationships that can be used to advantage in developing an effective solution procedure. After carefully considering the two contending representations from this point of view, we find the two NETFORMs based on the constructions of fig. 3 and fig. 5 equally well suited to our purposes. (In other settings, where a solution strategy of a different form may emerge as advisable, one or another of an AND node and an OR node construction may be preferable to the alternative.) In the present setting, both representations have the same connectivity and succeed in conveying the same fundamental perspective of the overall problem. Moreover, as will be seen, we may use this perspective as a basis for an approach that maintains all flows at integer values at each step. Under the assumption of integer-valued flows, the representations based on figs. 3 and 5 are equivalent. Thus, we now turn to characterizing the solution procedure devised for this problem, and will refer without loss of generality to fig. 4 as the foundation for its development.

4. Solution procedure

Organizing the solution procedure around the NETFORM of fig. 4 leads to several key observations. To begin, note that the integer flow arcs create a disconnecting set for the problem. Removing these arcs decomposes the problem into two disjoint networks. More specifically, if it were possible to know the right integer flows on the arcs of the form $(\hat{j}, j)$, then the rest of the problem could be solved optimally by solving two pure network problems. Accordingly, the driving philosophy of the solution strategy is to seek a way of identifying which of the $(j, \hat{j})$ arcs should receive unit flows, making use of the decomposition created by such flows as a means of carrying out the analysis.

To differentiate the two networks that result from the indicated decomposition, we call the left network the *Weapon Bundle Network* and the right network the *Target Network* (where these names refer to the node sets that are unique to each network). For any integer flows on the $(\hat{j}, j)$ arcs, these two networks become bipartite transportation networks, susceptible to highly efficient solution. We show how these transportation networks are created in the following discussion, which subdivides the overall solution approach into three phases.

PHASE I

The first phase of the solution strategy creates a *parameterized relaxation* of the target network, which does not assume specific integer flows on the $(\hat{j}, j)$ arcs. This relaxation has two goals: (1) to provide an evaluation to be used in determining which $(\hat{j}, j)$ arcs should carry positive flows, and (2) to derive a bound for the optimum objective function value for the overall problem. Both goals ultimately depend on coordinating the solution of the target network with that of the weapon bundle network. In the first phase, however, the target network is solved without reference to information derived from the weapon bundle network.

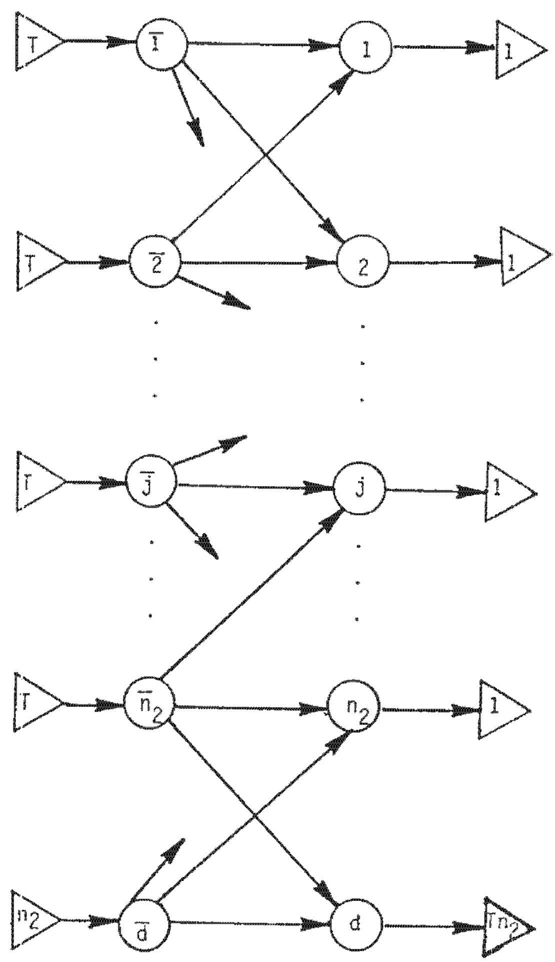

Fig. 6. Transportation structure of parameterized relaxation of the target network (costs not shown).

The transportation structure of the target network relaxation is shown in fig. 6. The network arises from the NETFORM of fig. 4 by the following steps. First, collapse the $(j, \bar{j})$ arcs of fig. 4, merging their two endpoints into the single node $\bar{j}$. We follow the convention of identifying nodes j, $\bar{j}$, and $\hat{j}$ by reference to their "common index" j. Give each node $\bar{j}$, for $j = 1, \ldots, n_2$, a supply of T units, and create two dummy nodes $\bar{d}$ and d, where $d = n_2 + 1$, creating associated zero cost arcs $(\bar{d}, j)$ and $(\bar{j}, d)$, for $j = 1, \ldots, n_2 + 1$. Each node j, $j = 1, \ldots, n_2$, is given a demand of 1 (dropping the single endpoint arc with a unit upper bound that emanates from this node), node $\bar{d}$ is given a supply of n_2 and node d is given a demand of Tn_2. Figure 6 depicts the structure of this network, disregarding costs.

For arcs other than the 0 cost arcs meeting nodes d and $\bar{d}$, there is a parameterization option. At the simplest level, if we employ the original costs (profits) C_{jk} for the arcs $(\bar{j}, k)$, as j and k range from 1 to n_2, then it is clear that the network of fig. 6 is indeed a valid relaxation, where the counterpart of this relaxation in the original problem is the "optimal transportation network" that results by the rules: (a) delete all nodes $\bar{j}$ from fig. 6 such that arc $(\hat{j}, j)$ of fig. 4 carries a 0 flow in an optimal solution; (b) eliminate nodes $\hat{d}$ and d and their incident arcs in fig. 6; and (c) compel a flow of 1 on each arc of the form $(\bar{j}, j)$ remaining. In particular, we may note that every feasible solution of the optimal transportation network translates into a feasible solution for the network of fig. 6 with identical cost.

The parameterization of fig. 6 is made possible because of the special status of arc $(\bar{j}, j)$, which must carry a unit flow if arc $(\hat{j}, j)$ carries a unit flow in the original problem. Thus, the costs of all arcs out of node $\bar{j}$ can be adjusted by replacing C_{jk} with C'_{jk}, where

$$C'_{jk} = C_{jk} + \theta_j/(T-1) \qquad k \neq j, \ k \neq d$$

$$C'_{jj} = C_{jj} - \theta_j \qquad j \neq d \ .$$

θ_j is an arbitrary parameter of any sign. This parameterization does not change the costs relative to the requirements of the original problem, since assigning a flow of 1 to arc $(\bar{j}, j)$ and a flow of 1 to each of $T - 1$ other arcs $(\bar{j}, k)$, $k \neq d$, achieves the same total cost regardless of the value of the parameter. The issue is to determine a value of θ_j that gives the strongest relaxation.

One approach is to use some variant of subgradient search, but the time required to execute the iterative adjustments of such a procedure would be undesirable relative to the overriding goal of solving the overall problem very rapidly. As a non-iterative option, a seemingly natural candidate value for θ_j is C_{jj}, since this gives all arcs $(\bar{j}, j)$ an "equal status" with a parameterized cost $C'_{jj} = 0$. However, reflection shows this will tend to produce a solution in which all $(\bar{j}, j)$ arcs have 0 flows. Instead we choose θ_j to be the value that causes C'_{jj} exactly to match the most profitable C'_{jk} for $k \neq j$. Experience shows this parameterization choice to be highly effective.

A still more important step in strengthening the use of the transportation network of fig. 6 occurs by conducting a post-optimality analysis of its optimal solution. This occurs at the conclusion of phase 1. The goal of this step is to identify the cost of assigning a unit flow to arc $(\hat{j}, j)$ in the original problem, using a measure based on determining the amount by which this flow assignment would cause the solution to the relaxed problem to deteriorate. For this, let π_j equal the reduced cost of arc $(\bar{j}, j)$ plus the sum of the $T - 1$ smallest reduced costs of those arcs $(\bar{j}, k)$ such that $k \neq j$ and $k \neq d$. (Such reduced costs are defined in the standard fashion [6,7] relative to the optimal network solution of the parameterized target network relaxation. Thus, these costs are all nonnegative and represent the marginal unit decrease in profit of sending flow on the associated arcs.) It follows that π_j is a valid lower bound on the deterioration in the optimal objective value for the target network relaxation caused by choosing $(\hat{j}, j)$ to receive a unit flow. Moreover, if J such arcs are selected, then the total deterioration of the relaxation is at least the sum of the J smallest π_j values. It may be noted from fig. 4 that $J \leqslant \min(n_1, n_2/T)$. The "correct" value of J, and an additional strengthening of this bound, will be described later.

Finally, we may also identify a cost δ_j that results from *not* giving a unit flow to $(\hat{j}, j)$. A legitimate value for δ_j equals T times the reduced costs of the arc $(\bar{j}, d)$, noting that the reduced cost will be 0 for each arc $(\bar{j}, d)$ that carries nonzero flow. The sum of the $n_2 - J$ smallest of the δ_j values also gives a bound on the total deterioration, though this bound turned out in practice to be dominated by the one based on the π_j values. The determination of the π_j and the δ_j values completes the execution of phase 1.

PHASE 2

The second phase of the procedure coordinates the target network relaxation with the weapon bundle network. This is done in a manner that utilizes both the π_j and δ_j values from phase 1, with the goal of bringing the two networks to a "consensus" on the right arcs to carry unit integer flows. At most n_1 arcs can be selected to receive such flows, and the weapon bundle network problem is solved in order to identify the best subsets of arcs that satisfy this restriction. (For our problem data, $n_1 < n_2/T$, and n_1 turns out to be a reasonably tight upper bound for J.)

The form of the weapon bundle network employed appears in fig. 7. Two dummy nodes e and $\hat{f}$, where $e = n_1 + 1$ and $f = n_2 + 1$, have been added to create the bipartite transportation structure. Each arc $(i, \hat{j})$ for $i \neq e$ and $j \neq f$ has a cost of π_j, while each arc $(e, \hat{j})$ for $j \neq f$ has a cost of δ_j. The cost of each arc $(i, \hat{f})$ for $i \neq e$ is 0, while the cost of the arc $(e, \hat{f})$ is $-M$, for M large. Thus, an optimal solution to this network will assign as many of the n_1 origins to the destinations as possible, and of these, the assignment will occur in such a way as to minimize the costs $\pi_j(\delta_j)$ of causing the relaxed solution for the target network to deteriorate as a result of selecting (not selecting) an arc $(\hat{j}, j)$ to receive flow.

Note that by subtracting δ_j from the cost of each arc into node $\hat{j}$, $\hat{j} \neq \hat{f}$, we obtain an equivalent problem where the new cost of every arc $(i, \hat{j})$ is $\pi_j - \delta_j$ except for the arc $(e, \hat{j})$, whose cost becomes 0. Because of the structure of this problem, a very effective advanced start is therefore obtained by ranking the $\hat{j}$ nodes in ascend-

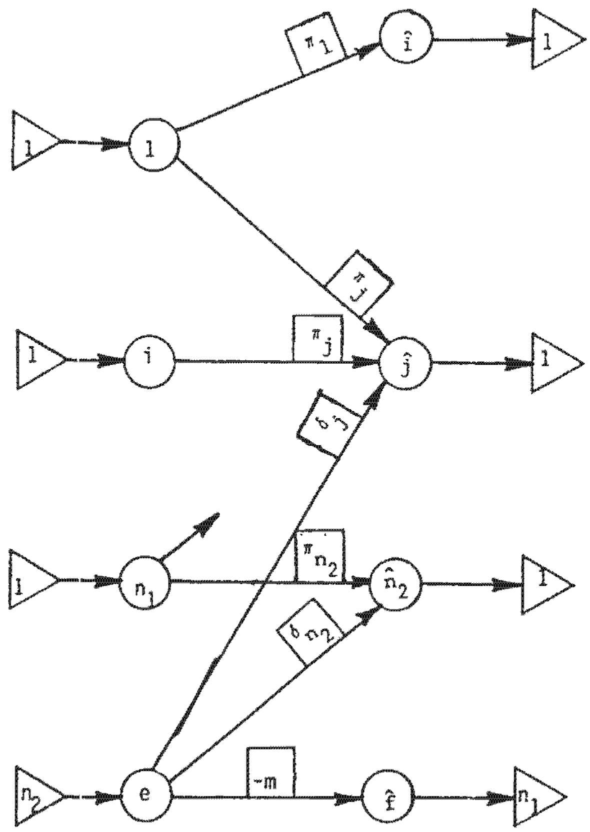

Fig. 7. Weapon bundle network to coordinate with target network relaxtion.

ing order of the $\pi_j - \delta_j$ values. The approach then assigns a unit flow to each $\hat{j}$ node in this ranked succession, choosing this flow to come from the node i, if one exists, that still retains its supply and connects to the smallest number of destinations not yet examined (subject to $\hat{j}$ being among them). The process stops when n_1 destinations receive flow, or all nodes $\hat{j} \neq \hat{f}$ have been examined, whereon remaining flows to achieve a feasible solution are assigned in the obvious way. This starting solution is frequently optimal for the weapon bundle network, and when it is not, the number of iterations to achieve optimality is extremely small. Thus, by the use of this start, the weapon bundle network requires very little computational effort. Phase 2 concludes by obtaining an optimal solution to this network.

PHASE 3

The final phase of the solution process uses the solution to the weapon bundle network from phase 2 as a basis for redefining the target network. In particular, each node $\hat{j}$ that receives a unit flow in the optimal weapon bundle network is assumed to identify correctly the associated node $\bar{j}$ of the target network which must be given a supply of T units. Thus, the target network is reduced to the form of an "optimal transportation network", as defined in phase 1, except that the dummy nodes $\bar{d}$ and d are retained to allow for the possibility of infeasible solutions. The provision for handling infeasibility when it occurs is simply to eliminate the supply of T units for the node $\bar{j}$ with the greatest associated π_j value, and then to post-optimize. In theory, the post-optimization step would be carried out as many times as necessary. In practice, however, the step was not required more than once, and typically was not required at all. (The relationship $TJ \leqslant Tn_1 < n_2$ undoubtedly contributed to this outcome.)

The transportation problem of this final phase, of size $J + 1$ by $n_2 + 1$ for $J \leqslant n_1$, is notably smaller than that of phase 1, enabling it to be solved more rapidly. The speed of solving both the original and final target network problems was further accelerated by an advanced start procedure. This procedure constitutes a direct generalization of the starting procedures for the weapon bundle network, implemented by sorting the profits of arcs into each destination node in descending order and then sorting the destination nodes as in the starting procedures of phase 2. In this case, however, the node sort is based on the value of the most profitable arc into the node, from an origin whose supply is not yet extinguished. (Origin node availabilities and hence the identities of "most profitable" existing arcs, are updated at each iteration.) The sorting and updating work was accomplished in negligible time due to the small integer range of applicable costs, permitting the use of an address calculation sort. (Note that the θ_j parameterization for the target network can effectively multiply this range by T, but the adjusted range still did not exceed 100.) As a result, the larger and the smaller target network problems were both solved very efficiently.

After obtaining the solution to the target network of phase 3, we undertook to generate a tighter optimality bound by means of a further refined relaxation. We did this solely as a means of testing how close our solution came to achieving the tighter bound, and did not incorporate the refined relaxation or its solution into the strategy for solving the problem. (An additional pass using this relaxation might be appropriate for more difficult problems where total solution time has a lower priority.)

The refined relaxation arises as follows. Note that for any choice of J a flow of $JT < n_2$ units must be sent from nodes $\bar{j}, \bar{j} \neq \bar{d}$ of fig. 6, to nodes other than d. This implies $n_2 - JT$ units must be sent from node $\bar{d}$ to these nodes, or equivalently, JT units can be sent on the arc $(\bar{d}, d)$. Thus, if this latter arc is given an explicit lower and upper bound of JT (effectively reducing the supply and demand values at its endpoints) and the π_j penalties are calculated as before, a tighter bound is obtained.

The value of J that results by solving the weapon bundle network is an upper bound on the optimal J, and therefore gives a natural "first J" to check. The approach was repeated for the next smaller value of J, but the resulting bound was invariably at least as tight as the first. Since the least restrictive bound obtained by stepwise checking is the one that is theoretically acceptable, we accepted the "first J" bound as the basis for estimating the proximity of our solution to true optimality. It is interesting to note that the calculation of these two optimality bounds generally took at least as much time as required to obtain the solution that these bounds were used to evaluate.

5. Computational results

Computational testing was applied to a set of benchmark problems whose statistics are shown in table 1. The costs (values of the targets) ranged from 1 to 10, and the arc multiplier T (number of missiles in a "bundle") was exactly 10 in all problems, independent of problem dimension and arc density.

Five different problems were solved for each of the seven sets of problem dimensions shown in table 1, thus constituting thirty-five problems in total. Specific problem data were supplied by the Strategic Air Command, and all runs were performed on an IBM 4381. The column indicating total number of variables is based on the most economical of the formulations, i.e. the one based on the construction of fig. 5. Thus, this number equals the sum of the Level 1 and Level 2 arcs, plus the number of Level 2 nodes (which gives the number of Z_{jj} variables). The Z_{jj} variables remove as many Level 2 arcs as there are Level 3 nodes, but this number is recovered in the bounded slack arcs associated with Level 3 nodes. The table reports average values for the objective functions and the percent of optimality bounds, but gives maximum computer CPU times (hence each problem of a given dimension was solved within at most the length of time specified).

The contribution of the NETFORM model in reducing the combinatorial complexity of the model is evident from the fact that the number of zero-one variables of the original integer formulation equals the number of Level 1 arcs plus the number of Level 2 arcs, which ranges from 106,900 zero-one variables to 984,000 zero-one variables, in contrast to the range from 1,200 to 2,400 zero-one variables for the NETFORM. Even with this combinatorial reduction, the problems are still extremely large, containing roughly a quarter of a million to nearly one million variables in total. Consequently, they pose a significant test for the efficacy of the insights derived from the NETFORM and embodied in our customized solution procedure.

As the table shows, the deviations from the optimality bounds obtained by this procedure were only a fraction of a percent, with smaller deviations for larger problems. The outcomes suggest that the application of NETFORM models and solution methods specially designed to exploit their structure can be a powerful tool for solving weapon deployment and targeting problems.

Table 1 *

Number of Level 1 nodes	Number of Level 2 nodes ☆	Total number of nodes	Number of Level 1 arcs †	Number of Level 2 arcs	Total number of variables	Average OBJ value	Average % upper bound on OPT value	Maximum CPU time (seconds)
100	1,200	4,900	102,000	144,000	247,200	8,970	99.6%	11.63
100	1,200	4,900	96,000	144,000	241,200	8,973	99.7%	11.40
100	1,200	4,900	102,000	288,000	391,200	8,994	99.9%	12.9
100	1,200	4,900	96,000	288,000	385,200	8,995	99.9%	13.08
100	1,500	6,100	127,500	225,000	354,000	8,994	99.9%	13.2
100	1,500	6,100	127,500	450,000	579,000	8,998	99.9%	15.8
200	2,400	9,800	408,000	576,000	986,400	17,990	99.9%	53.86

* Results for each problem size are the averages of five separate problems on an IBM 4381.

☆ Equals the number of Level 3 nodes and also the number of zero-one variables in the NETFORM.

† The sum of the number of Level 1 arcs plus the Level 2 arcs equals the number of zero-one variables in the original integer formulation.

References

[1] Advanced Missile Computer Model (AMM) Maintenance Manual, MM-07 Assignment subsystem, Logicon Report R:OAD:79003, 225 W. Fifth St., P.O. Box 471, San Pedro, CA 90733 (1979).

[2] J. Elam, F. Glover and D. Klingman, A strongly convergent primal simplex algorithm for generalized networks, Math. Oper. Res. 4, 1(1979)39.

[3] F. Glover, Transformations enlarging the network portion of a class of LP/embedded generalized networks, J. Information and Optimization Sciences 3(1982)196.

[4] F. Glover, J. Hultz and D. Klingman, Improved computer-based planning techniques, Part 1, Interfaces 8, 4(1978)16; and Improved computer-based planning techniques, Part II, Interfaces 9, 4(1979)12.

[5] F. Glover and J. Mulvey, Network relaxations and lower bounds for multiple choice problems, INFOR 20, 4(1982)385.

[6] P.A. Jensen and J.W. Barnes, *Network Flow Programming* (Wiley, New York, 1980).

[7] J.L. Kennington and R.V. Helgason, *Algorithms for Network Programming* (Wiley, New York, 1980).

[8] Capt. S. Monaco and D. Billings, Optimal targeting of ballistic missiles in a tiered aimpoint system, Report SRL-TR-78-005, Frank J. Seiler Research Laboratory (AFSC), U.S. Air Force Academy, CO 80840 (1978).