Appeared in Proceedings of Conference on AI & Logistics, 1990.

Ammunition Inventory Planning
An Integration of Configuration and Resource Allocation Techniques

*Carolyn Dunmire
U.S. Army Human Engineering Laboratory
(301) 278-5857

Neena Sathi, Rajay Goyal, Mark Fox, Alexander Kott
Carnegie Group Inc
(412) 642-6900

1. INTRODUCTION

The Inventory Planner is an interactive, decision support system for allocating an inventory of ammunition components into a desired set of complete rounds. It is being developed at Carnegie Group Inc by the Army Human Engineering Lab. The Inventory Planner is intended to support the logistician in analyzing the current ammunition stockage situation in terms of deliverable rounds meeting the mission requirements, rather than in terms of individually stocked components.

This research demonstrates how configuration planning and resource allocation planning can be combined so that the allocation of resources (components in inventory) will maximize the total number of requested configurations (mission-specific mixes of complete rounds), given one or more interchangeable parts (rounds or components of rounds). To properly manage inventories of ammunition components, possible complete round configurations and allocation of inventory to these configurations are two distinct problems which should be solved jointly. This will enable an logistician to make intelligent decisions on the options available to support the commander from limited inventories of ammunition components.

2. BACKGROUND

A "complete round" of ammunition comprises all components used in firing a weapon once, which, in the case of separate loading cannons, includes a projectile, propelling charge, primer, and fuze. The mission of a combat unit dictates the type of round or the component requirements for a complete round. For example, personnel targets and vehicle targets may demand the same high explosive (HE) projectiles, but hard vehicle targets will require impact fuzes, while soft targets in open areas may require timed fuzes for mid-air explosion.

For maximum tactical and supply flexibility, the Army encourages interchangeability of ammunition components. However, since a component can be used across multiple complete round types, the task of configuring one type of round impacts the ability to satisfy other types of rounds. This "competition" for components across round types increases the difficulty in determining how configuring one type of round will impact on future ability to configure other types of rounds, and prevents the logistician from making maximum use of limited but crucial ammunition inventories.

Currently, the only possible view of an ammunition inventory is from the level of components rather than from the level of complete rounds. From this view it is difficult to project which components are in short supply and how the components in inventory can be configured into the complete rounds that on operation is likely to require. No tool is currently available to analyze inventories from the view of complete rounds and answer questions such as:

- What is the maximum number of rounds of type X possible from this inventory?
- What components are compatible with what other components and with which weapons?

- What will be the inventory shortfall if the planned operation require certain types of rounds with a given composition?

The problem of inventory planning by complete rounds is twofold - determining valid combinations for a desired round type (terminal effect) and allocating the on-hand inventory to these valid round configurations.

3. PREVIOUS RESEARCH

Previous research has focused on either the configuration planning or resource allocation planning problems; each problem being solved independent of the other. This research is unique in that it addresses both the configuration and the resource allocation aspects in an integrated solution.

Given a pre-defined set of objects, configuration planning problems deals with finding a way of arranging the objects such that the final arrangement (configuration) satisfies a given set of requirements. Examples of configuration planning problems include: (1) configuring basic loads, (2) configuring convoy shipments, and (3) configuring the layout of facilities, and so forth. Many recent applications of AI to configuration appear to follow one of two approaches - the decompositional (abstract refinement model) [6, 7, 9] and the transformational model [3, 5, 8]. The decompositional model represents the configuration process as a sequence of steps, in which each step starts with an incomplete configuration state and produces a configuration state of greater completeness. The transformation model represents the configuration process as a sequence of steps, in which each step starts with a configuration state and produces another configuration state of the same degree of completeness by replacing a part of the configuration structure with a different sub-structure. The configurator module of the inventory planner is based on the general problem reduction (decompositional) configuration technology. For the present problem, the Configurator Module uses a practical and fast configuration enumerator based on constraint satisfaction, which provides a basis for developing more complicated configuration tasks in the future.

Resource allocation planning problems deal with the optimal allocation of resources across competing needs. Examples of resource allocation problems include the allocation of fires, the allocation of movement credits for a supply route among competing convoys, the allocation material handling equipment among competing jobs, and so forth. Previous work [1, 4] has focused on resource allocation and reallocation. The Resource Allocator Module of the Inventory Planner is based on this resource allocation work done by Sathi and Fox, and uses advanced constraint relaxation techniques as developed by Sathi.

4. MODE OF INTERACTION

The logistician establishes the problem to be solved by providing

- The supply locations whose inventories are to be included;
- The round types of interest ;
- The required number of rounds of each type; this can be expressed either;
  as relative proportions (each round relative to the others) or
  as an absolute number of rounds; and
- Optionally, the percentages of components types to use in a given round type
  (e.g., the percentage of MTSQ fuzes in 155 HE rounds);
  defaults are available for these proportions.

To aid in setting realistic goals, the logistician can request a "rough view" of the number of rounds of a given type available in a specified inventory. This number will indicate the maximum number of a round in the inventory disregarding the competition for components among round types. This rough estimate provides a ceiling for the number of rounds of a given type that the logistician can expect.

The Inventory Planner will then plan the total number of rounds of each type that can be assembled with the given inventory. Other reports will also be available, allowing the logistician to look at the results of planning from any of several perspectives. For example, a useful view of the inventory components "left over" (not allocated) after the planning task is completed is available. Leftover inventory may indicate a component inventory that is not in balance, components which are stocked but not in demand, or obsolete components which are not required for assumed future missions.

The Inventory Planner allows the logistician to accomplish several tasks in a single session. For example, the logistician may want to see how well the same set of goals can be satisfied with different inventories or compare how well a single inventory can satisfy several different sets of goals. By making small changes in the problem setup, the logistician can play "What-if" games, for example, ask how many more 155-mm HE rounds can be configured if he decreases the requirement for 155-mm APERS rounds. The logistician can also generate unrelated plans for various sets of supply locations, each with its own set of goals. Finally, the logistician can save various parts of a plan or goal specification to a file for later use.

5. TECHNICAL APPROACH

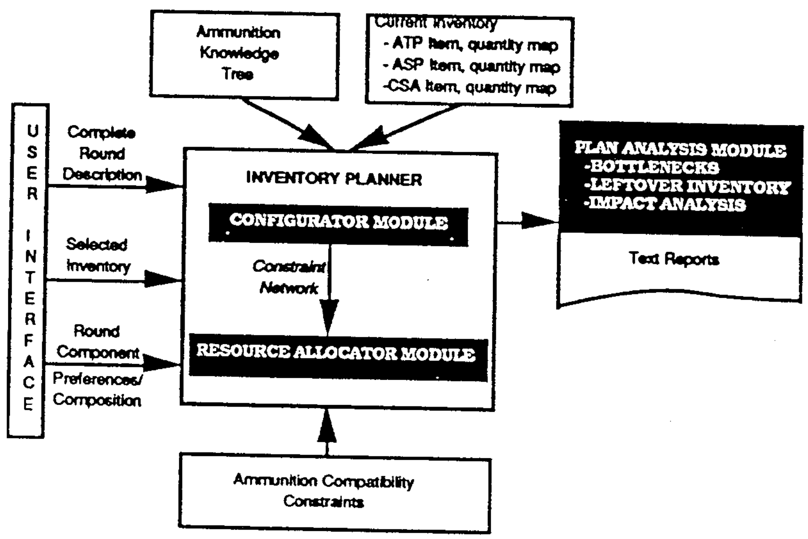


Figure 1 Inventory Planner Architecture

As shown in Figure 1, there are three major modules of the Inventory Planner:

- **Configurator Module**
- **Resource Allocator Module**
- **Analysis Module**

The Configurator Module creates all feasible configurations of a given round based on compatibility constraints. This tree of possible configurations is used by the Resource Allocator Module to determine actual rounds to be assembled.

The Resource Allocator Module assigns a quantity to each feasible configuration based on available component inventories and on quantity and composition constraints on the weapons, rounds, and/or components of a given inventory or combination of inventories.

The Analysis Module provides analysis of the plan and suggests which components are in short supply (bottlenecks). It is also used for explaining the analysis of the left over inventory that could not be used in any of the requested configurations.

5.1 CONFIGURATOR MODULE

Configuration knowledge about component interchangeability and compatibility with weapons is stored in a hierarchical tree structure of frames called a configuration knowledge tree. The Configurator refers to the configuration knowledge tree for each round to be configured and finds what alternate choices are available and how to subdivide the task of configuring a complete round into smaller tasks of configuring its subcomponents.

Another source of constraints are the user-defined preferences on round type, component type or component model number input during the problem setup. Finally, the total number of components at a given supply location also impacts the total number of rounds that can be configured from that inventory. These availability constraints are used by the Configurator Module to prune the search space of any components not available in inventory, and are used by the Resource Allocator Module to determine component shortfalls.

Problem-Solving Method

As described in Figure 2, the Configurator Module executes a two-step process:

Step 1 Generate Configuration Types from User Requirements.

Step 2 Create all valid feasible configurations from configuration types, where each configuration description includes a projectile model, a fuze model, a propelling charge model, and a primer model.

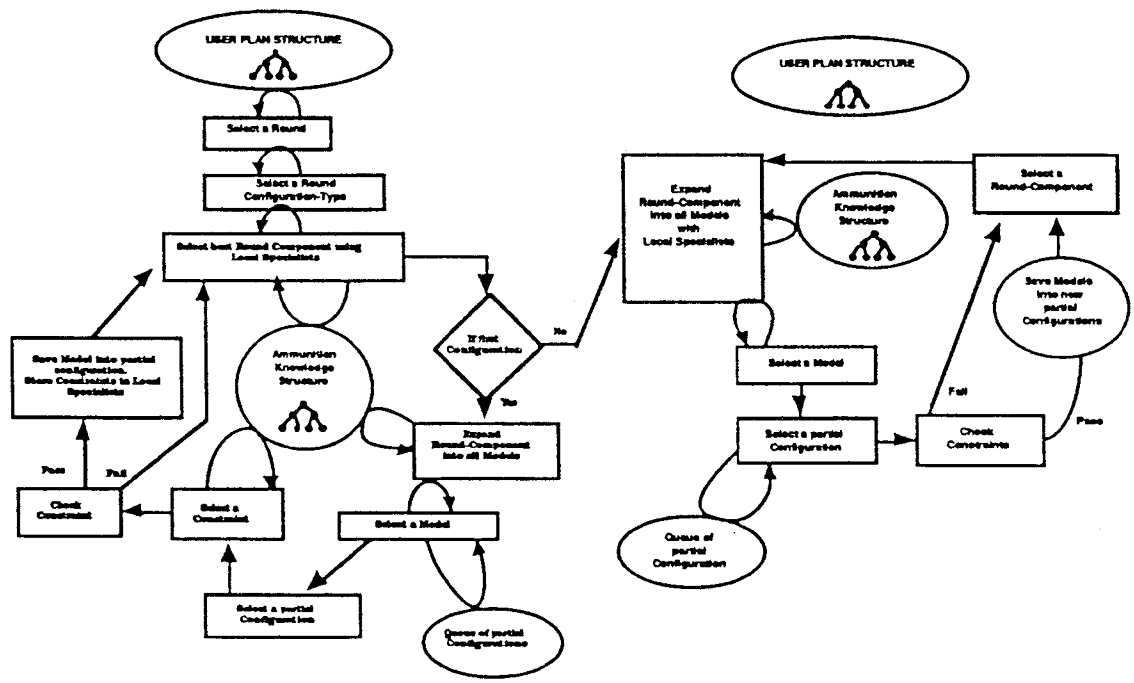


Figure 2 Algorithm for Configurator Module

- ***Generate Configuration Types.***

The purpose of this step of the configurator solution is to create all possible component combinations for a given round. Each component type specified by the user is combined with all other component types to create the set of all possible combinations of fuze, primer, projectile, and propelling charge for this problem. For example, consider a hypothetical case that has three round components, namely, projectile HE-MK171, fuze PD, and fuze MTSQ. The possible combinations satisfying the restriction would be simply (projectile HE-MK171, fuze PD) and (projectile HE-MK171, fuze MTSQ).

Any components not specified by the user will default to the complete set of available, compatible components of that type (e.g. all fuzes that are currently stocked). At the end of this stage all created combinations of round components will be stored as configuration types with links to the round components selected and to the round, thus adding one more level to the plan structure.

- ***Create Feasible Configurations.***

This stage of the Configurator solution process expands the listing into all the model level combinations that satisfy the compatibility-only constraints contained in the configuration knowledge tree. This stage relies upon the use of Local Specialists to improve the efficiency of the search and pruning process. A key performance issue is determining the best order in which to match the round components needed for the round in question. A round component containing constraints that match many other components is best to use as a first round component to match. Encoded Local Specialists attached to every round type give us this information.

The round component is expanded into all its possible model number level elements. For every model number level element, we check all constraints attached to that model. A constraint that matches a given partial configuration is one in which all model elements in the partial configuration are acceptable. If the partial configuration contains a fuze, projectile, and primer and a particular constraint only limits the primer, the other components do not impact the search. The compatibility-only constraints guarantee that only those elements will be saved that can potentially lead to valid configurations. The model level elements are stored in a queue of partial configurations.

The next round component is selected to be configured. Then all partial configurations are expanded by adding in these model level elements. For every new partial configuration, the constraints are checked for the newly added model. Only those partial configurations that pass are saved. This is repeated until there are no more round components to be matched. When this happens, the list of surviving partial configurations becomes the list of feasible configurations for the given user-specified round. The surviving configurations are stored as configurations, thus adding one more level to the logistician plan structure.

Performance

Different test cases of increasing complexity were developed to validate the performance of the Inventory Planner. The objective of the first test case was to configure the maximum total number of two different complete rounds while keeping the percentage relationship constant. The objective of the second test case was to configure the maximum number of total rounds. In the third test case, six rounds had a relative relationship with each other and two rounds had a specified quantity as a goal. The planning objective was to maximize the total number of rounds while meeting the quantity of the two rounds and the percentage relationship between the other six rounds. The table below illustrates the performance of the Configurator Module with these test cases.

Performance Statistics on the Configurator Module

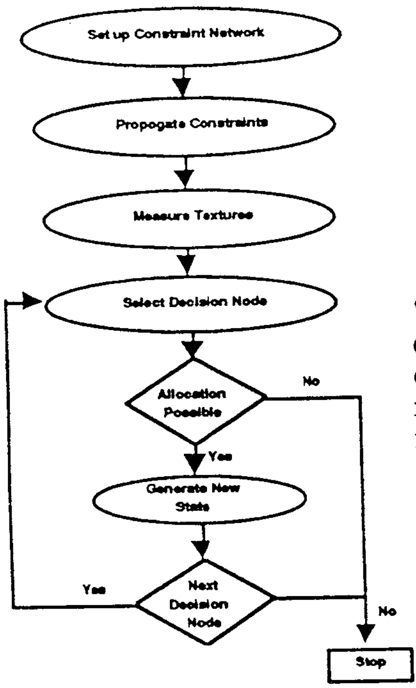
Test	Search Size	Number of Rounds	Number of Configurations	Symbolics 3640 CPU-Time
1	15,000	2	63	9.14 secs
2	60,000	3	100	16.85 secs
3	200,000	8	198	16.31 secs

As can be seen, the CPU time does not have a monotonic relationship with the number of configurations. The CPU time taken depends on the complexity of the underlying compatibility constraints and number of constraints.

5.2 RESOURCE ALLOCATOR MODULE

Determining the availability of complete rounds can be viewed as a resource allocation problem; each acceptable configuration of a complete round requires the conjunctive allocation of one or more of each of a projectile, fuze, primer and propellant. Optimizing the set of complete rounds depends on the ability of the problem solver to allocate bottleneck components strategically. This approach to maximizing total number of order configurations is based on "Constrained Heuristic Search" (CHS) [2].

Problem-Solving Method



In CHS, the problem space is defined by a network of resource requirements imposed by the feasible round configurations. Problem solving proceeds by selecting a resource that is in high demand and allocating that resource to the configuration that relies upon its allocation the most. Figure 3 describes the algorithm for Resource Allocator Module, which has five major submodules:

- *Create Constraint Network* : The Configurator Module is called to set up all feasible configurations. Figure 4 represents a constraint network composed of nodes representing complete rounds as defined by the logistician, feasible configurations of rounds generated by the Configurator Module, and the ammunition component "resources" to be allocated by the Inventory Planner.

Once the constraint network is constructed, CHS is performed. The initial state in the search space contains the constraint network developed by the Configurator Module.

- *Propagate Constraints*

- **Calculate Component Demand:** For each component, determine the demand by summing the demand each configuration has for the component. A configuration's demand for a component is calculated by taking the product of its expected quantity and the number of component required per configuration.
- **Calculate Max-Rounds:** For each configuration, determine the maximum configurations possible, unconstrained by other round definitions.

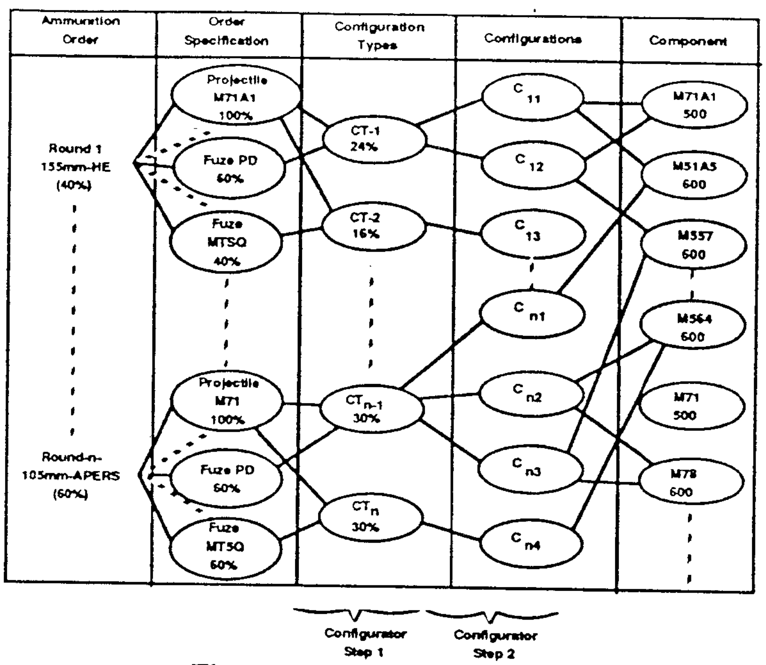


Figure 4 Constraint Network

- *Measure Textures*
  - **Calculate Component Contention:** For each component, determine the amount of contention by dividing demand for each component by available quantity. The greater the number the greater the contention.
  - **Calculate Configuration Reliance:** The reliance a configuration has on a particular order is function of number of feasible configurations of that particular order.
- ***Select Decision Node.*** Some of the heuristics for selecting the decision node include
  - Select the component with the greatest contention;
  - If there is more than one component with the same contention, select the one with smaller number of configurations; and
  - For a given component, select the configuration with the greatest reliance.
- ***Generate State.*** Once a component is selected, a new state is generated in which the quantity is allocated to the chosen configuration. The following changes are made in the constraint network in the new state:
  - Decrement component node's quantity by amount allocated.
  - Append to the component nodes allocation the tuple (<configuration>, <amount-allocated>).
  - Set the configuration node's amount _allocated to the specified amount.
  - Assign and allocate other component to the chosen configuration node.

There are cases when application of the constraints reveals that none of the alternatives can be used. This means that either the logistician requirements over-constrained the problem so that no solution is possible, or that some bad choices have been made earlier in the solution process. (In the case of over-constraint, the constraint can be relaxed by looking into maximum component demand and max-rounds for each configuration.) For this reason, it is necessary for the system to record each decision and its alternatives, so that dependency directed backtracking can occur.

Performance

The test cases used to validate the Configurator Module were also used to validate the Resource Allocator Module. The Configurator Module output configurations were used as input to the Resource Allocator Module. The performance of the Resource Allocator Module with these test cases is:

Performance Statistics on Resource Allocator Module

Test	Search Size	Number of Rounds	Number of Configurations	Number of Constraints	Symbolics 3640 CPU-Time
1	15,000	2	84	508	3.4 secs
2	60,000	3	100	590	8.6 secs
3	200,000	8	198	1026	11.3 secs

5.3 ANALYSIS MODULE

After the Inventory Planner has been run, data can be collected for analysis and presentation to the logistician. The Analysis Module is used for

- Reporting results of the planning process,
- Providing analysis of bottleneck components,
- Providing analysis of the left-over inventory, and
- Replanning by selective violation of constraints.

"What-if" questions can be handled by changing the parameters in the constraint network and then executing the Planner again.

6. FUTURE DIRECTIONS

The algorithms were developed in several iterations. The latest version of the algorithm incorporates dependency directed backtracking and chooses the resource to be allocated next based on both contention and reliance. Future versions will extend functionality to national stock number, Department Of Defense Ammunition Code, and Lot Number levels of specification and planning, and will incorporate mission level constraints for mission-based round substitution and determination of combat configured loads. The Inventory Planner was developed on a Symbolics computer, but it is currently being ported to the Sun/4.

For more information, contact Carolyn Dunmire, U.S. Army Human Engineering Laboratory, DSN 298-5857, 301-278-5857, net address cdunmire@hel4.brl.mil.

7. REFERENCES

1. Fox, Mark S. (1983). *Constraint-Directed Search: A Case Study of Job-Shop Scheduling*, Computer Science Department, Carnegie-Mellon University, Pittsburgh, Pennsylvania 15213.

2. Fox, Mark S., Sadeh, Norman, & Baykan, Can (1989). *Constrained Heuristic Search* (Abstract), Robotics Institute and Computer Science Department, Carnegie Mellon University, Pittsburgh, Pennsylvania 15213.

3. Howe, A., Cohen, P., Dixon, J., and Simmons, M. (1986). Dominic: A Domain-Independent Program for Mechanical Engineering Design. In Sriram, D., and Adey, R. (Eds), *Applications of Artificial Intelligence in Engineering Problems*. Berlin: Springer-Verlag. pp. 289-300.

4. Kung, N. and Yung, D.Y.Y. (1989).A Planning/Scheduling Methodology for the Constrained Resource Problem, *Planning, Scheduling, Reasoning About Actions*. Page 998-1003.

5. Kott, A.S., May, J.H., and Hwang, C.C. (1987). An Autonomous Designer of Thermal Energy Systems. *International Conference on Systems, Man, and Cybernetics*. Washington, D.C.: IEEE

6. Maher, M.L. (1984). HI-RISE: *A Knowledge-Based Expert Systems for the Preliminary Structural Design of High Rise Building*. Doctoral dissertation, Department of Civil Engineering, Carnegie-Mellon University.

7. Mittal. S., and Araya, A. (1986). A Knowledge-Based Framework for Design. *Fifth National Conference on Artificial Intelligence*. Philadelphia, PA.: CAI-86.

8. Murthy, S.S., and Addanki, S. (1987). PROMPT: An Innovative Design Tool. *Sixth National Conference on Artificial Intelligence*. Seattle, WA: AAAI.

9. Steinberg, L.I. (1987). Design as Refinement Plus Constraint Propagation: The VEXED Experience. *AAAI Sixth National Conference on Artificial Intelligence*. Seattle, WA: AAAI.