Appeared in: Proceedings of the DND Workshop on Knowledge-Based Systems and Robotics, Ottawa, 1993

Distribution Planning: A Constrained Heuristic Search Approach

Victor Saks, Ivan Johnson
Carnegie Group, Inc., 5 PPG Place , Pittsburgh, PA 15222
tel: +1-412-642-6900; fax: +1-412-642-6906; internet: saks@cgi.com

Mark S. Fox
Department of Industrial Engineering, University of Toronto,
4 Taddle Creek Road, Toronto, Ontario M5S 1A4
tel: +1-416-978-6823; fax: +1-416-978-3453; internet: msf@ie.utoronto.ca

Abstract

KBLPS is a state-of-the-art Decision Support System with a rich interactive graphics user interface, advanced knowledge base and powerful planning algorithms that has been designed and implemented to support skilled U.S. Army logisticians to prepare and evaluate logistics plans to support corps-level battle scenarios much more rapidly than currently possible. At the heart of KBLPS is the commodity Distribution Planner (DP) algorithm, which in a single framework supports Ammunition Distribution and Petroleum Distribution planning. KBLPS uses the problem solving model of Constrained Heuristic Search (CHS). CHS views the construction of the distribution plan as a constraint optimization problem and uses graph textures to guide and control the problem solving. KBLPS is fully implemented in C++ and has undergone a significant unit and integration testing, and is currently being used by the USArmy.

1.0 Introduction

The complexity of military logistics planning and emerging needs for ever-higher degrees of responsiveness motivate the requirement for automated decision support tools that complement the strengths of the human logistics planner. The Knowledge Based Logistics Planning Shell (KBLPS) under development by Carnegie Group, Inc. (CGI) and LB&M Associates with the sponsorship of the Combat Service Support Division of the U.S. Army's Human Engineering Laboratory (HEL) and the U.S. Army Strategic Logistics Agency (SLA) provides such support tools.

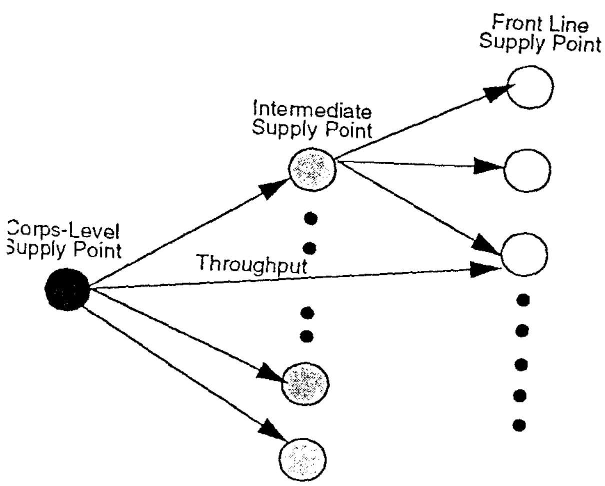
At the heart of KBLPS is the Distribution Planner (DP) planning algorithm, which underpins the Ammunition Distribution Planner (ADP) and the Petroleum-Oil-Lubricant (POL) Distribution Planner (PDP). A major challenge has been to design a single common algorithm general enough for use by these two domain-specific planners. The ADP and PDP support CORPS/G4 Cell planners in assessing the supportability of proposed Maneuver Courses of Action (MCoA) from ammunition and POL logistics perspectives. The DP algorithm reasons and calculates at appropriate levels of aggregation to demonstrate the logistics supportability of a MCoA without overwhelming the user with too much detail.

KBLPS and the Logistician can work together to generate commodity distribution plans much more rapidly and with greater accuracy, efficiency and responsiveness than now possible for the Logistician working alone. The Logistician can configure the problem/scenario and give guidance to the algorithm by optionally setting various parameters; the algorithm constructs a distribution plan which involves significant computational complexity. As a consequence, the Logistician can spend more time analyzing and assessing plans than generating them.

In the following, we describe the logistics planning domain in more detail, followed by a description of the problem solving method based on Constrained Heuristic Search [Fox et al., 1989]. Lastly, we describe the development process, observations and current status.

2.0 Problem Description

The DP addresses the problem of supporting a corps-level combat mission maneuver course of action by planning to move as much product as possible forward to the forward supply points, to meet the user-unit product consumption/demand over a user-specified time period (typically 3-5 days in this domain).



The above picture depicts the flow of materiel. Ammunition distribution requires the planning of ammunition shipments from Corps Support Areas (CSA's) to Ammunition Supply and Transfer Points (ASP's & ATP's); POL distribution plans POL shipments from General and Direct Support (GS, DS) points to Main and Forward Support Battalions (MSB's & FSB's). Each domain has particular aspects which make it complex. For example, there are far more Ammunition types (called DODICs) than POL product types. On the other hand, POL distribution resources such as tankers and storage bags are product specific, which is not the case for ammunition. The shape of the distribution network is potentially more complicated for POL.

The principal goal of distribution planning is to maximize the satisfaction of unit demand. Priorities must be respected so that higher priority orders are satisfied before lower priority orders. But the preference for higher priority orders must be tempered by the objective that lower priority orders cannot be completely ignored and starved; hence, there must be a balance between these conflicting goals.

An important secondary objective is to meet day-to-day inventory stockage objectives, which are Logistician-specified in terms of number of days of supply in the context of the CoA being supported. When in conflict with the primary objective, this objective is relaxed.

The distribution planner must assign Material Handling Equipment (MHE), Main Supply Routes (MSR), truck and available inventories to supply the user units consumption/demands in the best possible way over time. In general, this is an "over-constrained" problem - the net demand is generally substantially greater than inventory and/or material handling and carrying capacity can handle, at least in some time-frames. Hence, some of the demand will remain unsatisfied. It is the G4 Log Planner's responsibility, to find the means to move more material and/or to set aside the subset of demand that can best be "left aside" and still meet the commander's intent with the subset of materials that can be moved.

Moreover, in different scenarios different resources will be most scarce. Thus, the planner must maximize efficient usage of all resources, and in particular the scarce resources, in order to maximize over-all order satisfaction. This must be done in light of commander intent/mission, balanced servicing of all combat units so they can adequately support the mission and each other, and available assets.

3.0 Overview of Approach

The planning algorithm plans commodities distribution through its model of the distribution network of supply points, with the objective to maximize on-time satisfaction of demand, in the form of orders, and meet mission-driven stockage objectives. The generated plans are feasible - resource capacity constraints are not exceeded at the level of aggregation being used.

The DP algorithm constructs a plan by iteratively selecting an order (the 'next best' order not yet in the plan) and then placing that order into the plan. The DP reasons and plans "opportunistically", in the sense that neither the sequence in which orders are planned nor the way in which orders will be planned are determined in advance. Since there are typically a large number of orders which could be planned next and numerous ways to plan them, the DP uses *knowledge of the problem structure* to guide the search to make these decisions quickly, efficiently, and accurately. More precisely, high priority orders which rely on resources which are heavily contended for in certain time intervals are planned as early as possible in the planning process, to ensure that these high priority orders are not left out of the plan. The algorithm strives to relieve contention for over-constrained resources by planning elsewhere (in time and/or location) orders that are contending for those resources and that also have reasonable alternatives.

The DP algorithm provides a mechanism which allows a Logistician-specified percentage of higher priority orders to be satisfied first, then a percentage of lower priority orders, and then the remainder of the higher priority orders. In this way, priority is respected but lower priority units and orders are not starved completely. This balance mechanism gives control to the Logistician, and is consistent with the KBLPS philosophy of being a Decision Support System.

3.1 Constrained Heuristics Search

KBLPS uses the problem-solving model of Constrained Heuristic Search (CHS) [Fox et al., 1989]. CHS combines the power of Constraint Satisfaction (CSP) techniques with heuristic search. It uses a constraint graph to model a problem and uses measurements of the properties of the constraint graph, which we call textures, to guide the search process of iteratively selecting a variable and assigning a value to it.

CHS models a logistical problem using a temporal/capacity constraint graph (figure 1). The nodes are variables to which one or more values are to be assigned. In the logistics domain, nodes in the graph represent activities to be scheduled, i.e., assigned a start time, end time, and a set of resources. The arrows represent temporal constraints between activities that the assignment of the start and end times must satisfy (e.g., precedence constraints). In this case we show only "before" temporal constraints. The heavy arcs are capacity constraints. They are disequalities that specify that two connected activities cannot use the same resource at the same time, i.e., cannot be assigned the same resources during the same temporal period.

An important component of the CHS model is that there exist domain independent properties of the constraint graph, called textures, that can be used to construct search heuristics:

Value Contention: degree of to which more than one variable wish to have the same value.

Variable Reliance: degree to which a variable relies upon the assignment of a particular value.

Variable Looseness: size of range (conjunction of constraints).

Constraint Tightness: degree to which the constraint reduces the set of admissable solutions.

Constraint Importance: how important is it to satisfy the constraint.

Figure 2 depicts a *Contention Graph*, which is a elaboration of the temporal/capacity constraint graph. It adds an additional node for each resource so that texture measures can be stored in the graph at the resource nodes. The picture depicts the aggregate demand over time that exists for Resource R_3 as imparted by the individual demands of Activities A_{13}, A_{21}, and A_{23}

Problem solving is performed by iteratively selecting an activity and assigning a temporal period, i.e., start time and end time, and one or more resources. In particular, the following steps are followed cyclically until all activities have been assigned resources and time

Figure 1.

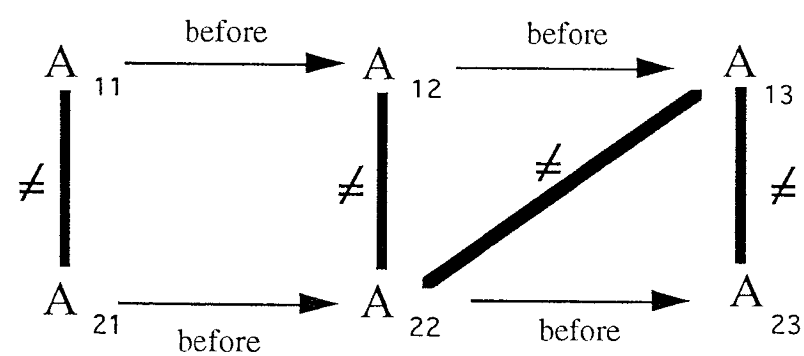
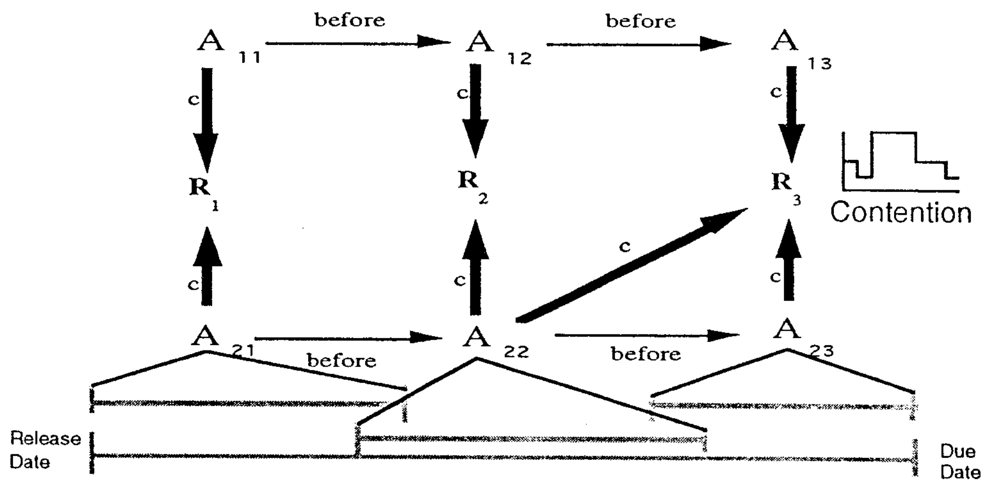


Figure 2.



periods. Given a logistics problem modelled using a contention graph:

1. Perform constraint propagation (aka arc-consistency).
2. Measure textures.
3. Select an activity.
4. Assign a temporal period and resources.

In the context of distribution planning, textures that are measured are contention for a resource from activities, and the reliance an activity has on a particular resource. The DP algorithm first identifies which resources are most heavily in demand relative to their availability at specific time intervals within the planning horizon, and then assigns them to the most reliant activities so as to exploit them as completely as possible.

Backtracking occurs whenever a deadend is reached. Experience with this approach in the domains of factory scheduling and spatial planning have shown that backtracking is minimal [Sadeh, 91; Baykan & Fox, 92].

3.2 Algorithmic Steps

KBLPS implements a variation of CHS based on the work of [Sadeh & Fox, 90; Sadeh 91]. For each order for ammunication or POL, there exists more than one way of satisfying it. A chain of activities defines the 'process steps' that must be performed to move material forward; these include truck loading, hauling a quantity of commodity from one supply point to another, etc. Each order can be satisfied by more than one chain of activities, resulting in a tree of alternative activity sequences. Activity Model 'templates' define these generic process/activity sequences. The union of these activity tree templates for each order defines the constraint graph for the distribution planner.

At this point, the constraint graph is not yet complete. The demand an individual activity has for a resource is not yet known in the activity tree template for an order. The demand for ammunition (or bulk petroleum) is probablistically propagated through the distribution network, thereby distributing an individual order's demand across the alternative activity chains that may satisfy it.

With the constraint graph now complete, textures can be measured. The impact on the modeled resources (trucks, roads, MHE) if the demand were to be totally satisfied is calculated. The need for the resources is compared to their availability in 'time/resource chunks'. The resource(s) most in demand relative to their availability are singled out for scrutiny. The demand (in the form of back-propagated orders) contending for those resources are reviewed for criticality; those orders most dependent (they have fewest alternative ways of being satisfied) on those resources are selected to be placed into the plan. This is done in such a way, however, (e.g., by 'sliding' them in time) as to relieve the bottleneck effect on the heavily-contended resource(s) as much as possible. The remaining (as yet un-scheduled) part of the problem is then re-assessed and the next-most-bottlenecked resources and most-critical orders are dealt with. This cyclical process continues until there is no remaining demand to satisfy or there are no time/resource fragments remaining that can be used to build another plan segment.

Of course, the process is considerably more complex in detail than this summary might suggest. For example, at the first step, propagation of demand through the distribution network, a probabilistic view is imposed, recognizing that there is uncertainty in exactly when material movements will actually be assigned, how long shipment will take, which orders and associated quantities will succeed in capturing resources, etc.

The basic sequence as the Plan is generated is: (1) Measure Textures; (2) Select an Order; (3) Plan the Order; (4) Record the plan fragment. We discuss these steps here in more detail:

1. *Measure Textures:* For each activity, determine its probablistic demand for each resource it needs, such as ammunition, material handling equipment, etc. For each resource, aggregate the probablistic demand from each activity that needs it. For each activity, determine its reliance on each resource it needs.
2. *Select Order*: Select the (approximately) most-contended-for resource in some specific time interval. At this point a few or many resources may be approximately equal in their contention measure. Being able to control how approximate this equality measure is gives us the capability to trade off some detailed precision for execution speed; the algorithm can select one of the approximately most-contended for resources much more rapidly than selecting the exactly most-contended for resource. (An underlying observation here is that the contention measures are probabilistic, and hence not exactly precise.) We are continuing to explore how sensitive plan results and computation speed are to these approximation factors. Numerous researchers in planning and scheduling have discussed order selection. We have been influenced most by [Fox et al., 1989; Sadeh & Fox, 1990; Sadeh, 1991]

   The DP algorithm then selects an order from the group of propagated orders contending for the selected highly-contended resource at that time. This selected order has approximately the highest "Reliance" on the critical resource in that time interval. High Reliance reflects the order's need to be planned on the critical resource in that time interval; from a planning perspective, it is important to make that decision now (i.e., on this plan fragment cycle) before the critical resource is consumed by another order which doesn't need it as badly. The algorithm also works to plan the order without using the critical resource in the critical time interval, thereby helping to relieve that bottleneck. Moreover, the specific activity (e.g., unload trucks) performed on the critical resource (e.g., limited MHE) is the focus for planning the order; we refer to this as the "critical activity" (CA).
3. *Plan the Selected Order*: Planning the CA is the focus of planning the order. First, the algorithm constructs a feasible time window for the CA. This is done by propagating backwards through the order's Activity Model from the order due time and forwards from the earliest feasible time for beginning the shipment (time is represented continuously). The CA can be laid into any (sufficiently long) time block within this window; the order will be feasible with respect to all of its other activities.

   The algorithm works to choose the start time for the critical activity which reduces the contention for the critical resource to the maximum extent possible, using the contention measures discussed above. The algorithm then proceeds to plan up- and down-stream activities for the order with a just-in-time approach. Each order is planned such that its execution is feasible; hence, the plan is feasible at each successive step (cycle) of the plan-building process; at whatever point the data drives the algorithm to stop processing, the plan will be feasible. There may be some demand that remains unsatisfied as resources have been consumed by higher priority demand, but the overall plan as generated will be feasible according to resource availabilities.
4. *Record Plan*: Having laid another plan fragment into the overall emerging plan, the algorithm updates the resource consumption/availability tables and the demand profiles for resources which have been affected. Some constraint propagation is also performed at this point as needed; e.g., when resources become exhausted in some time interval, demand for those resources is recalculated for the time intervals where they have not been totally consumed.

4.0 Development Process

The developmental process has gone through several stages of approach and intensity during early requirements definition and more recent focused development:

- Conceptual definition of logistics planner tasks, and what subset could be reduced to engineered software solution(s); this was accompanied by a series of early (LISP-based) prototypes to help clarify requirements, needs, and possibilities; these efforts led to first versions of the DP algorithm outside of a basic research environment.
- First fully integrated KBLPS version, using scanned paper maps, limited (notional Xth corps) battle scenarios; expanded second version (applied research) of DP algorithm; still largely LISP-based environment.
- Current fully integrated (fully C/C++) version; software engineering methodology (using CASE tools) to develop firm design requirements, coding and testing of GUI, KB, and DP and Transportation Scheduler algorithms; step-wise unit testing and module integration; exhaustive stand-alone algorithm testing, refinements, and changes to increase execution speed; exhaustive integrated testing; current on-going field testing with skilled world-class logisticians.

  The current version of KBLPS has been tested on problems of a maximize size of 4,000 orders, 20,000 activities, and 300 resources. The average execution time for a problem with 2,000 orders and 10,000 activities is 2.5 seconds on a Sparc10.

5.0 Validation Process

In order to facilitate testing and validation, we developed a comprehensive Test-Case Generator (TCG), which takes a high level scenario description as input and outputs a full scenario description which can be input into the DP. We first generated small simple scenarios, gradually upgrading the TCG so that it now can build scenarios that stress the algorithm in specific and controllable ways. The TCG was indispensable as we debugged and improved the DP; we now have a test suite of over 100 test cases.

It is useful to distinguish between bugs that cause the system to crash and bugs which do not. The former are usually much easier to find, isolate the cause, and repair. The DP algorithm is complex; it was inevitable that there will have been design and coding errors which do not cause the system to crash, but do cause the algorithm to make poor decisions. These bugs are much harder to find, isolate the cause, and fix. After a substantial baseline algorithm evaluation and debugging process was completed, the DP was reliable in terms of running to completion even on completely new scenarios. We then spent a second significant amount of time examining DP results for plan quality, i.e., looking for situations where the DP made a poor decision, figuring out why, and improving the algorithm to make a better decision in that situation. Unfortunately there really is no alternative to this labor intensive method.

The algorithm subsequently went through a thorough examination by the domain experts at LB&M Associates. As a result, they were quite impressed with the algorithm's responsiveness and sensitivity to changes in the model, so that, for example, if a little more resource is added, the algorithm predictably plans more orders, and vice versa.

The result of this painstaking debugging process is that the DP is quite robust, running successfully to completion on new scenarios and generating high quality plans.

6.0 Observations

The design of the DP algorithm has been challenged by the need to consider carefully the trade-off between plan quality and execution speed. Too much calculation can take too long, while too little can result in lower quality plans. This issue has been noted by other researchers trying to build a real system with users in the field [Sadeh, 91; Zweben et al., 90]. As we gain more experience with the algorithm's behavior and the utility of the plans it produces, we anticipate design refinements to continue to balance speed vs. quality. One aspect may well be other layers of aggregating data (e.g., placing groups of ammunition types into a single "demand bucket" thereby reducing the data-intensity of a problem scenario.)

We have succeeded in designing a single algorithm capable of handling both Petroleum and Ammunition commodity distribution. There are some key differences between these, perhaps most notably the heavy reliance on Throughput (direct ammunition trans-shipment from CSA to ATP) and less reliance on Throughput with Class III. KBLPS enables the logistician to specify the mix of throughput and supply point-to-supply point shipment he prefers to aim for in any particular scenario. The DP algorithm respects those user-specified preferences and works toward satisfying orders accordingly.

The DP algorithm is designed to be opportunistic, so that while it is constructing the plan, its selection of the most constrained resource on which it focuses its attention is completely dynamic. For each of the modeled resources, we have test case scenarios in which that resource is clearly the most constrained, and the DP continuously focuses its attention on that resource. In other scenarios, the are several resources which have heavy contention and the DP shifts its focus from one to the other, so that as it finishes planning one bottle-necked resource, it shifts its attention to another.

The DP is also designed to relieve bottlenecks, so that sometimes as the DP actually relieves a bottlenecked resource by planning orders elsewhere, that resource becomes less bottlenecked and the DP shifts its attention to a more bottlenecked resource. Shifting of attention also occurs with respect to time, so that the DP can maintain its focus on one resource but shift the focus to different times throughout the planning horizon. All of these shiftings of focus contribute to the construction of a high quality robust plan, in which heavily contended-for resources are planned to be utilized with near maximal efficiency.

7.0 Status

KBLPS is currently being used by the XVIIIth Airborne Corps, Fort Bragg, to plan logistical support for rapid deployment. KBLPS has also been chosen by the US Army Materiel Command as one of the sustainability tools for the "Lousiana Maneuvers". It is also being acquired for use in Europe, Korea and by the Joint Chiefs of Staff J4.

KBLPS is being integrated into the curriculum at the US Command and General Staff College, Fort Leavenworth, Kanas, for training logisticians in distribution planning. It is also being integrated into the Army War College, Carlisle PA, general officers' command post.

8.0 Summary

The Knowledge Based Logistics Planning Shell (KBLPS) is a cutting-edge application combining the best of conventional (X-windows, Motif, etc.) and Artificial Intelligence (Knowledge Base, constraint-directed scheduling algorithms) technologies. There is substantial potential for further growth and development, including adding new applications modules platformed on top of the rich knowledge base already in place. KBLPS represents an emerging capability to accomplish fast and accurate logistics planning at several echelons in the army.

This application is currently undergoing intense evaluation and use by skilled logisticians, supporting their on-going efforts to plan for regional contingencies around the world. The user's, and their senior management's, current enthusiasm and support bode well for further constructive development, extensions, and deployments in the near future.

9.0 Bibliography

[Baykan & Fox, 92] Baykan, C.A., and Fox, M.S., (1992), "WRIGHT: A Constraint Based Spatial Layout System", in *Artificial Intelligence in Engineering Design, Volume I*, Tong, C., and Sriram, D. (Eds.), Orlando Fl: Academic Press.

[Fox et al., 89] M. S. Fox, N. Sadeh, C. Baykan. Constrained Heuristic Search. In Proceedings IJCAI-89, 1989

[Sadeh & Fox, 90] N. Sadeh and M. S. Fox. Variable and Value Ordering Heuristics for Activity-Based Job-Shop Scheduling. In Proceedings of the Fourth International Conference on Expert Systems in Production and Operations Management, 1990.

[Sadeh, 91] N Sadeh. Look-Ahead Techniques for Micro-Opportunistic Job Shop Scheduling, Ph. D. Thesis, CMU, 1991.

[Zweben et al., 90] M. Zweben, M. Dale, R. Gargon. Anytime Rescheduling. In Proceedings of the DARPA Workshop on Innovative Approaches to Planning and Scheduling, 1990.