Proceedings of the 1992 Winter Simulation Conference
ed. J. J. Swain, D. Goldsman, R. C. Crain, and J. R. Wilson

ISLE -- INTELLIGENT SCALABLE LOGISTICS ENVIRONMENT

Lisa Sokol
Steven Geyer
Robert Lasken
Katherine Murphy

MRJ, Inc.
10455 White Granite Drive
Oakton, Va 22124, U.S.A.

ABSTRACT

This paper presents the proposed architecture for ISLE, an intelligent Scalable Logistics Environment. ISLE incorporates three different solution techniques: simulation, AI and OR in a massively parallel environment. ISLE is based on data parallelism. ISLE supports an incremental technique for scheduling. It can represent illegal schedules, e.g., schedules whose tasks require an over commitment of resources, or whose task schedules do not meet temporal constraints. ISLE uses of both local and global optimization strategies.

1 INTRODUCTION

Logistics simulation models are large and complex. The size of the search space and the combinatorial possibilities for component assignment require the use of intelligence to render the computations feasible. We propose the development of an intelligent data parallel (Hatcher, 1991) to support the analysis of deployment schedules. Experimental results confirm that data-parallel programs can achieve high speedups on MIMD computers. ISLE is an iterative approach to scheduling in support of logistic systems. The primary advantage of iterative schedules is that when constraints change, the scheduling process is not restarted from scratch. In many environments, constraints may change during the implementation process. Each iteration will focus in part on bringing the schedule back to a feasible region. ISLE allows the representation of illegal schedules, e.g., schedules whose tasks require an over commitment of resources, or whose task schedules do not meet temporal constraints. Allowing illegal schedules to briefly exist increases the amount of parallelism inherent in the application. It also simplifies the use of parallelism by allowing simultaneous changes to be made to the schedule without requiring each change to fully "understand" its global impact. These simultaneous changes will on occasion form an illegal solution that will need to be repaired in a later iteration. The schedule is refined and conflicts are resolved using a meta-level strategy tool.

2 BACKGROUND

2.1 Background: Massively Parallel Deployment Model

MRJ first developed a massively parallel deployment scheduling simulation for the U.S. Army Concepts Analysis Agency (CAA). The simulation, which runs on our 16,384 processor Connection Machine (CM2), integrates inter and intra-theater deployment simulations into a single seamless, detailed model. The simulation emulates the features from two existing CAA models. The first, called TRANSMO, covers the inter-theater and includes scheduling, loading and routing of both air and sea vehicles from CONUS and other ports of embarkation to the theater. The second, called SITAP, covers the intra-theater and includes port resources for handling and storing packages, and scheduling, routing and loading of helicopter, railroad and wheeled and tracked vehicles into the intra-theater. The code develops routes, plans itineraries, and allocates lift assets to execute the plan. The basic objects in the problem are lift assets, networks, nodes and packages. Events occur as a package moves through the simulation. The events are first planned and later carried out. Packages are assigned path through the network that will bring the package to its ultimate destination at approximately the desired time. The packages are processed through a complex set of rules. The primary rule for scheduling is the arrival of packages at their ultimate port of debarkation on or about their latest arrival date (LAD). As the simulation proceeds, various packages compete for assets to assist in their movement.

The current model runs in approximately seven minutes for the Northeast Asia Scenario. The original model took approximately two days of computer and human time. This time is for a 16,384 physical processor Connection Machine virtualized eight times to a 131,072 virtual processor machine. (Virtualization occurs when each connection machine processor is split into more than one "virtual" processor in that each processor emulates more than one processor.) The processing encompasses the movement of 993 movement requirements from CONUS and other ports of embarkation to the Korean theater and into the Korean theater to ultimate ports of debarkation. The movement requirements spawned approximately 40,000 package events. In the simulation module, these package events were completed, and they spawned approximately 80,000 vehicle events.

2.2 Background: Data Parallel

A data parallel approach has the following characteristics: imperative style, explicit, local view of computation, and emphasis on loosely synchronous execution of single instruction stream (Hatcher 1991). Data parallel applications have five very desirable attributes: versatility, practicality, programmability, portability or relative machine independence, and finally reasonable performance. Experimental results confirm that data-parallel programs can achieve high speedups on MIMD computers.

2.3 Background: CLIPS

We have recently completed a project which created a data parallel software version of CLIPS version 5 system (Giarrantano, 1991) developed by Johnson Space Center. CLIPS was designed and built in the 1980s to enable expert system capabilities to be integrated with more conventional software systems. The current release of CLIPS includes a rule-based development environments, a functional programming language and a powerful object-oriented programming system.

It is our belief that parallel CLIPS will create a good support environment for ISLE. Compared to any system we may build from scratch, CLIPS is a mature system. It already supports a user community and is actively being used in the development of intelligent systems. NASA makes the C source code available allowing it to be ported to wide range of machines and supporting user modification. Since CLIPS was designed with modification and integration in mind, it is possible to integrate it with a DBMS capability and to extend CLIPS for data parallelism.

We made three modifications to earlier versions of CLIPS to create a support environment. The first was to modify CLIPS syntax so that it can support important data parallel code features. Secondly, CLIPS was modified to provide support access database management systems. This will allow ISLE to connect to existing databases, encourage database use throughout execution, and support the parallel rule-based component. Basic to understanding how this will be accomplished is to recognize the similarity between rule patterns and relational database queries. The facts found in CLIPS are equivalent to records found in a database and for each possible rule pattern that CLIPS can describe, there is an equivalent relational database query that would be able to find the facts desired by the rule. The query language that we designed is SQL like. The actual queries are executed on a CM2. Finally, CLIPS was modified so that it can get parallelism out of rules by converting parallel facts into databases and using parallel database accesses to supply parallelism. During a set of benchmarks, we found that to process one million facts on an 8K processor Connection Machine-2A took 4.55 seconds. In a Sun based sequential computing environment it took us seven seconds to process two thousand facts.

2.4 Lessons Learned

In the process of implementing MPDA, we gained valuable lessons that could significantly enhance a second effort. The first lesson was that writing the heuristic knowledge and database queries in a functional language (Paris) made later system modification more difficult. Second, the preplanned nature of the simulation (which was consistent with the original CAA deployment planners) made it difficult to modify the simulation for a more dynamic environment. Third, the original system was not designed with portability in mind. As technology has progressed it has become evident that software portability, even in a parallel environment, is a desirable design characteristic.

We believe that a knowledge-based system implemented in a supercomputing environment will help ameliorate the problems associated with the size and the complexity of the logistics scheduling domain, the level of detail required, the satisfying multiple conflicting goals and the need to be reactive. The rule-based software will be used to support dynamic simulation execution. The rule-based software will also make the simulation more flexible and adaptable. In another unrelated project, MRJ has developed a data parallel knowledge-based technology which we believe could be used to support a parallel knowledge-based system.

3 "INTELLIGENT SCALABLE LOGISTICS ENVIRONMENT" - ISLE

The most common approach to schedule construction is to build a conflict-free schedule by adding one task at a time. If an infeasible schedule is reached, the traditional constructor backtracks until feasibility is reached. ISLE takes a different approach. It is an iterative approach to scheduling in support of logistic systems. The primary advantage of iterative schedules is that when constraints change, the scheduling process is not restarted. Each iteration focuses in part on bringing the schedule back to a feasible region. ISLE supports the representation of illegal schedules, e.g., schedules whose tasks require an over commitment of resources, or whose task schedules do not meet temporal constraints. The concurrent changes made to ISLE during an iteration will occasionally form an illegal solution which require repair in later iterations. The schedule is refined and conflicts are resolved using a meta-level strategy tool. Biefeld (1991) discusses a sequential approach to scheduling which supports iterative scheduling, illegal schedule definition, and a multi-level search hierarchy.

The two primary functions of this tool are situation assessment using information about the state of the schedule from prior iterations and strategy selection. Situation assessment is the process of determining how well the current schedule meets the system goals, relative to the scenario defined system constraints and the system measures of effectiveness (MOEs). During each iteration, the meta-level strategy tool searches for bottlenecks, overuse of resources, etc. The tool must capable of differentiation between local versus global optima. The tool must also be able to discern when it appears that further iterations will no longer improve the solution. Finally, the high-level strategy tool must determine the focus for the more local levels of strategy. These areas selected for increased focus are the inputs for a second set of local assignment strategy heuristics. It is the role of this next heuristic level to identify possible actions and to determine parameters. These tasks are concerned only with local optimization. ISLE is also non-traditional because it uses different strategies for different portions of the scheduling problem. For example, shallow types of search are used when there are many tasks to examine, and deep search is used when there are limited numbers of tasks to consider There are four basic components to ISLE: scenario generation, global assessment, local assessment strategy, task assignment as (see Figure 1).

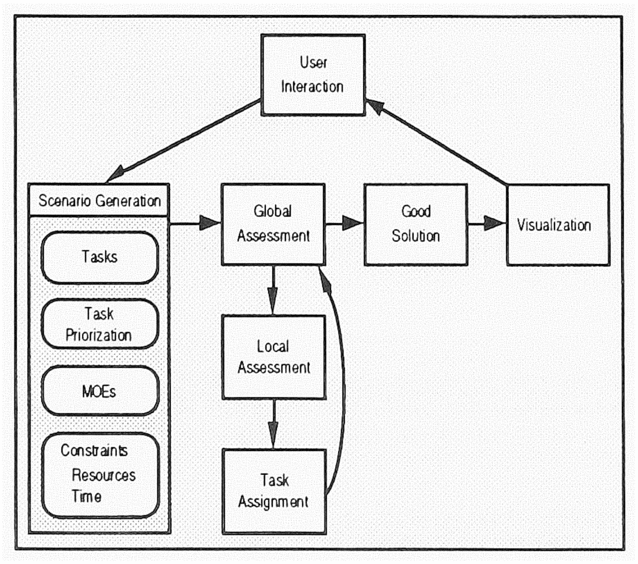

Figure 1: ISLE Overview

3.1 Scenario Generation

The deployment problem includes multicommodity and multi-fleet aspects, scheduling to arrive in specified time windows, variable environment, nonlinear cost functions, and large numbers of nodes, links, vehicle, and commodities. There are four basic components to the deployment model: task definition, task prioritization, constraints, and measures of effectiveness. Task prioritization measures the importance associated with task execution. Measures of effectiveness (MOE) are used to evaluate resource allocation decisions. Some candidate MOEs include: percentage of packages arriving at their ultimate port of debarkation on or about their latest arrival date (LAD), percentage of high priority tasks executed, amount of traffic congestion, unit cohesiveness, and vehicle utilization rates.

The basic components with ISLE are lift assets, networks, nodes and packages, as shown in Figure 2. The network is comprised of a series of links between ports. The packages consist of cargo and people. The cargo is categorized by type. We expect to extract the general data concerning cargo characteristics, geographic locations, port characteristics, etc., from the TPFDD (Time Phased Force Deployment Databases). These objects are tied together through movement requirements which specify earliest-start, latest-start, earliest-finish, minimum duration, maximum duration and end locations and times for a package. The constraints that we anticipate incorporating in the model include: allocating port resources for handling and storing packages; and scheduling, routing and loading of helicopter, railroad, and wheel and track vehicles in the intra-theater. One can design the algorithms which consider additional factors such as vehicle quantity, capacity, and speed as they relate to package urgency. Additional rules accounting for path vulnerability, unit cohesiveness, and improved vehicle utilization would also possible with minor modification because ISLE is extensible.

3.2 Global Assessment Strategies

ISLE is an iterative approach to scheduling in support of logistic systems. The schedule is refined and conflicts are resolved using a meta-level strategy tool. The two primary functions of the meta-level strategy tool include situation assessment using information about the state of the schedule from prior iterations and strategy selection. Situation assessment means that the strategy tool must assess how well the current schedule meets the goals of the system as given by the system scenario. There are two sides to schedule improvement. The first is an increase in the value of the schedule relative to the measures of effectiveness. The second is a decrease in the infeasibility of the schedule. During each iteration, the meta-level strategy tool must assess the system for bottlenecks, overuse of resources, etc. The tool must ensure that the system does not get caught in local optima and must be able to determine when ISLE has developed its best schedule. Finally, the high level strategy tool must determine the focus for the more local levels of strategy.

The driving force behind logistics models is the size and the complexity of the large search space associated with large deployment models. The size of the search space and the combinatorial possibilities for component assignment require the use of intelligence to render the computations feasible. We propose to combine earlier work conducted by Biefeld (1991) with two somewhat similar approaches intelligent search heuristics, tabu search (Glover, 1989) and decision-theoretic techniques (Hansson, 1990), to the global assessment strategy portion of the problem. Each of these meta-level tools offer the possibility of incorporating intelligence into the search space and determining areas of the schedule which require further modification. We propose to implement this model in a supercomputing environment. First a short background on each heuristic.

Tabu search is a strategy for solving combinatorial optimization problems. It is an adaptive procedure with the ability to make use of other heuristics. The primary function of tabu search is to overcome the limitations of local optimality within most search heuristics. There are two key elements to tabu search: that of constraining the search by classifying certain of its moves as forbidden (i.e., tabu), and that of freeing the search by a short-term memory function that provides "strategic forgetting." There is a dual relationship between the tabu restrictions and the aspiration criteria as a means for constraining and guiding the search process, and introducing the use of intermediate and long-term memory functions that operate in counter point to the function of short-term memory. Tabu restrictions allow a move to be regarded as admissible if they do not apply, while aspiration criteria allow a move to be regarded as admissible if they do. This complementarity of the underlying tabu restrictions and aspiration criteria enables them to be integrated into a common framework.

Decision-theoretic search (DTS) is a recently developed search technique that uses probabilistic decision theory to optimize the use of heuristic metrics during a search. It offers a sounder method for limiting the overall search space than classic approaches such as A*. It offers more than monotonicity and admissibility in search techniques. DTS is valuable for three reasons. It formalizes how heuristics information is used during search. It adapts to the search in progress. Finally, it can be implemented in a parallel environment

3.3 Integration of Tabu Search and Decision-Theoretic Heuristic

The Decision-Theoretic Search (DTS) and tabu search integrate well. DTS can build upon the most simple and atomic measures of advancement such as a priority scheme. We will substitute the tabu rules for the simpler and more explainable DTS search rules to gain the documented benefits of tabu search. We will use tabu's rules about the next node to be expanded and the straight forward evaluation procedures of DTS to determine which of tabu's rules to use. As a result, we should achieve speed within an evaluative framework, DTS, and will be able to characterize which heuristics to apply next.

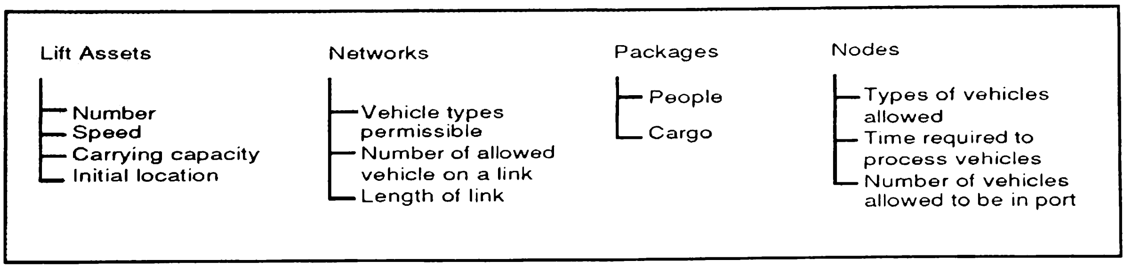
Lift Assets
Number
Speed
Carrying capacity
Initial location
Networks
Vehicle types permissible
Number of allowed vehicle on a link
Length of link
Packages
People
Cargo
Nodes
Types of vehicles allowed
Time required to process vehicles
Number of vehicles allowed to be in port

Figure 2: Object Attributes

There are actually three components to meta-level strategy assessment. The short-term memory component of tabu search, is handled by treating the tabu list as a circular list. It is this tabu list which prevents immediate circularity in the search space. The next two components to the tabu heuristic are rule-based and will be incorporated into the DTS. The first of these rule-based components will deal with regional intensification of the search, while the second will play the role of global diversification within the search for feasible strategies. When these two components are combined with the short-term memory functions fulfilled by the tabu lists, they provide an interplay between learning and unlearning. The regional intensification portion will record and compare features of a selected number of best-trial schedules developed during a particular period of search. Features that are common to a majority of these solutions are taken to be a regional attributes of good solutions. The method then seeks new solutions that exhibit these features, by correspondingly restricting or penalizing available moves during a subsequent period of regional search modification.

The role of the global diversification portion of the strategy is roughly the reverse of those for regional intensification portion. Instead of inducing search to focus more intensively on regions that contain good solutions previously found, the function of the global diversification is to guide the process to regions that markedly contrast with those examined thus far. The approach differs from those methods that seek diversity by generating a series of random starting points and hence which afford no opportunity to learn from the past. The objective is to create evaluation criteria that can be used by a heuristic search process which is specifically designed to produce a new starting point by purposeful instead of random means. These evaluation criteria penalize the features that long-term memory finds to be prevalent in previous solutions.

One of the major aspects of tabu search derives from induced search behavior that compounds the types of patterns produced by a single tabu list. One of these effects of such a tabu list is to create a succession of solutions in which the objective function value oscillates. This will be advantageous in applications such as these to define feasible solution and to uncover opportunities for improvement that are not readily attainable when the search is narrowly confined. Additionally, an osciliating strategy will permit the discovery of good solutions for perturbed conditions, where constraints may be slightly relaxed or tightened. Knowledge of solutions that are inside and outside feasibility boundaries are useful because they can provide insight as to the robustness of a particular schedule. Search termination occurs when the increase in the solution falls below some threshold or the search space becomes empty.

3.4 Local Assignment Strategies

These areas for increased focus are the inputs for a second set of local assignment strategy heuristics. It is the role of this next heuristic level to identify possible actions and to determine parameters. These tasks are only concerned with local optimization. We propose the combination two different approaches, multi-agent reasoning and tabu search. The multi-agent reasoning will be used in ISLE to focus attention in particular areas of the schedule and to make modifications in these areas. The meta-level heuristics will set "policy" for these low level agents by changing both the rules and the facts used by the agents. Both controls are necessary to promote efficiencies during the search process. The rules represent the "how facts are combined". The facts are used to keep the state of the simulation in a readily accessible form. ISLE will use the agents to limit the scope of planning, just as we limited our local planning. These agents will be given simple local heuristics and reduced fact sets from the current state of the schedule. Two advantages of this approach is that it focuses the computation in the local areas requiring attention and it is able to be evaluated very rapidly.

ISLE will use the perspective of each agent to structure their domain expertise. As in the previous discussion of the tabu search, we will use three levels of heuristics to support the agents. The first will be a circular tabu list. The other two levels of heuristics will contain rules supporting regional intensification of the search, and global diversification within the search for feasible strategies. As a result some agents will reshuffle the temporal aspects of a plan, some will change only the assignments of a class of resources, some will do pair-wise transforms on semantically similar resources, and some will be priority fixed. By controlling where computational resources are utilized, this approach provides either a more rapid answer or a more "informed" one. ISLE can use the multiple agent perspective to vary the local results also. This increased variation can break the local planning out of local optima, and lead to results similar to other "noisy" algorithms such as simulated annealing. Classical modal approaches, implemented with simple propositional constructs, will be our starting point. A parallel rule-based system based on CLIPS will be the support mechanism for this approach.

3.5 Task Assignment

The final level actually carries out the instructions of the independent local assignment strategy heuristics. This is the re-assignment of tasks to resources. As noted earlier, the use of locally based-assignment schedules can cause infeasible schedules to be generated. Each of these sets of task reassignments can be executed in parallel because schedule infeasibility is allowable. On the completion of this final task, the meta-level strategy assessment task is called once again to assess the state of the schedule.

4 REFERENCES

E. Biefeld and L. Cooper, August 1991. Automated Scheduling Via Artificial Intelligence. *NASA Tech Brief*, Vol. 15, No. 8, item #8, August 1991, JPL Invention Report NPO-18209/7721.

S. Geyer. August 1990. CLIPS Meets the Connection Machine or How to Create a Parallel Production System. *Proceedings of the First CLIPS User's Conference*. Houston.

J. Giarrantano. January 1991. *CLIPS User's Guide*, CLIPS Version 5.0, NASA Johnson Space Center, Information Systems Directorate, disseminated by COSMIC.

F. Glover. Summer 1989. Tabu Search - Part 1. *ORSA Journal on Computing*, 1:3:190.

O. Hansson. August 1990.Decision-Theoretic Planning in BPS. *Proceedings of the 1990 AAAI Symposium on Planning.*

P. Hatcher. 1991. *Data-Parallel Programming on MIMD Computers*, Massachusetts Institute of Technology, MIT Press, Cambridge, Mass.

AUTHOR BIBLIOGRAPHIES

LISA SOKOL received her Ph.D. from University of Massachusetts in 1978. Dr. Sokol is a senior member of MRJ Inc.'s Supercomputing Division. Her primary interest is parallel/distributed decision envrionments.

STEVEN L. GEYER received is B.S. in Electrical Engineering, University of Illinois. He is currently a senior member of MRJ Inc.'s Supercomputing Division. Mr. Geyer is interested in the use of intelligent decision techniques to applications such as large database sysems, and image processing systems. Mr. Geyer is also responsible for the design and first implementation of the Parallel Production System (PPS).

ROBERT LASKEN received his Ph.D in Physics, from the University of Maryland in 1972. Dr. Lasken is interested in the application of algorithmic and optimization techniques to applications in parallel hardware environments. He also received special award from the Gordon Bell Competition for implementation of a nonlinear network optimization algorithm on the Connection Machine.

KATHARINE MURPHY received her BSE from Arizona State University in 1987 and her M.S. from George Mason University in 1992. She is a member of the technical staff at MRJ Inc. Her current areas of interest include parallel algorithms and applications for database systems.