DAS

Intelligent Scheduling Systems for Shipbuilding

*Jae Kyu Lee, Kyoung Jun Lee, June Seok Hong, Wooju Kim,
Eun Young Kim, Soo Yeoul Choi, Ho Dong Kim,
Ok Ryul Yang, and Hyung Rim Choi*

■ Daewoo Shipbuilding Company, one of the largest shipbuilders in the world, has experienced a great deal of trouble with the planning and scheduling of its production process. To solve the problems, from 1991 to 1993, Korea Advanced Institute of Science and Technology (KAIST) and Daewoo jointly conducted the Daewoo Shipbuilding Scheduling (das) Project. To integrate the scheduling expert systems for shipbuilding, we used a hierarchical scheduling architecture. To automate the dynamic spatial layout of objects in various areas of the shipyard, we developed spatial scheduling expert systems. For reliable estimation of person-hour requirements, we implemented the neural network–based person-hour estimator. In addition, we developed the paneled-block assembly shop scheduler and the long-range production planner. For this large-scale project, we devised a phased development strategy consisting of three phases: (1) vision revelation, (2) data-dependent realization, and (3) prospective enhancement. The DAS systems were successfully launched in January 1994 and are actively being used as indispensable systems in the shipyard, resulting in significant improvement in productivity and visible and positive effects in many areas.

Daewoo Shipbuilding Company is one of the largest shipbuilders in the world. Its main products are the very large crude oil carrier (VLCC) and the container ship. Other products include heavy machinery, plants, and automobiles; these products are minor contributors to the company's sales. Its shipbuilding yard has three docks: One is the largest dock in the world, capable of manufacturing a 1-million-ton VLCC, that has a Goliath crane that can hold as much as 900 tons. There are 12,000 employees, and the sales in 1993 were $2 billion.

Shipbuilding is make-to-order manufacturing, and it takes a long time to complete the products. There are many complicated processes between an order and the final delivery, and a large amount of working capital is required. It is difficult to manage the material, human, facility, and information resources. Therefore, the scheduling and control of a shipbuilding plant is a very complex task—a nightmare.

The Company's Struggle with Scheduling

The company has struggled since its conception for an effective scheduling process to achieve total optimization. Poor scheduling keeps workers waiting for the prerequisite subassemblies; causes fluctuations in work loads, resulting in expensive overtime work; and causes delays in delivery. The company tried various project management programs, such as PROJACS, VISION, and X-PERT, as well as in-house development using conventional programs, but these efforts were unsuccessful because the flexibility of the scheduler's knowledge, as well as complicated interrelated factors, could not be represented. In addition, no software could support the dynamic spatial layout even though the spatial resources associated with material-handling equipment such as cranes are bottlenecked. To develop the integrated scheduling expert systems that could overcome these problems, the Korea Advanced Institute of Science and Technology (KAIST) and Daewoo jointly led the Daewoo Shipbuilding Scheduling (DAS) Expert System Project from 1991 to 1993. The DAS systems were successfully launched in January 1994 and have been become indispensable systems in the shipyard, resulting in a significant improvement in productivity and positive effects in many areas.

Application Description

The DAS Project adopted various expert system technologies to cope with the scheduling task. Key subsystems in DAS are (1) DAS-ERECT, the erection scheduler at docks; (2) DAS-CURVE, the curved-block assembly shop scheduler; (3) DAS-PANEL, the paneled-block assembly shop scheduler; (4) DAS-MH, the neural network–based man-hour estimator; and (5) DAS-LPP, the long-term production planner.

DAS-ERECT: Erection Scheduler at Docks

The initial shipbuilding process begins with block division. Typically, a large tanker is divided into hundreds of various-sized blocks, as shown in figure 1. These blocks can be categorized into two types: (1) flat bottom, which are assembled in the paneled-block assembly shop (PBS), and (2) curved bottom, which are assembled in the curved-block assembly shop (CBS). Multiple blocks might be assembled into larger ones (called *superblocks*) in the preerection shop (PES) to reduce the assembly time needed at the main erection dock. At the main docks, a number of blocks and superblocks are constructed ("erected") using a large Goliath crane. In a shipbuilding plant, typically one or two dry docks are for erection, with corresponding preerection shops, as shown in figure 2.

The important characteristics of erection scheduling are as follows:

Sequential Erection at Each Dock: Blocks and superblocks in a dock have to be erected

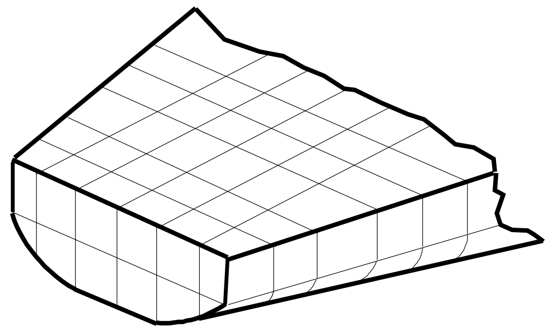


Figure 1. A Ship and Its Partition into Blocks.

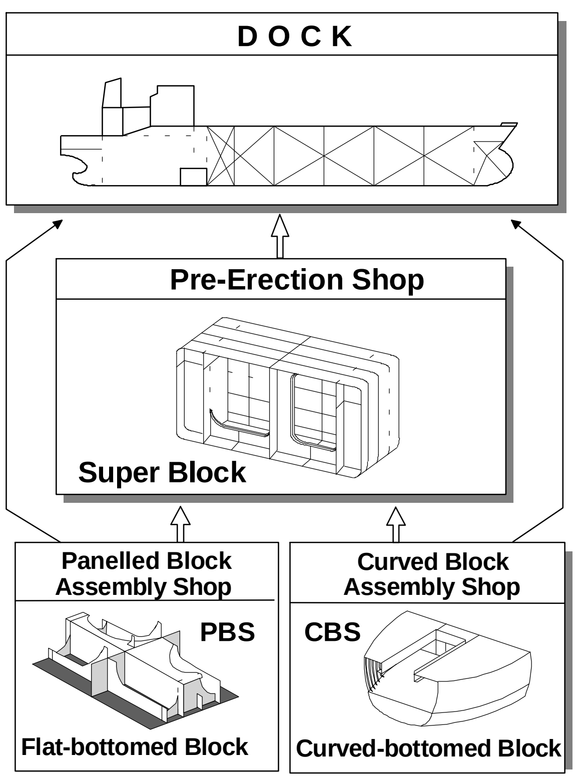


Figure 2. Typical Shipbuilding Procedure.

one at a time because there is only one Goliath crane at each dock. However, some plants might have multiple docks.

Large Search Space: Because the schedulers have to consider multiple ships, each composed of 400 to 500 blocks, a manual search for the best schedule is beyond the mental-processing capacity of any human scheduler. Thus, there is a need to develop a proper search method.

Technical Knowledge for Erection Sequencing: The technical knowledge that restricts the erection sequences is available.

Use of Resources: Key resources in shipbuilding include human work forces, cranes, and space. To use these resources effectively, a schedule should be established to balance the loads among the PBS, the CBS, and the preerection shops and docks.

Hierarchical System Architecture for Shipbuilding Scheduling

For erection scheduling at the dock, we used a hierarchical scheduling architecture (Lee, Suh, and Fox 1993) and developed the constraint-directed graph search technique (Lee et al. 1994). In the hierarchical architecture, detailed schedules for the lower-level assembly plants are delegated to the individual plant's schedulers (DAS-CURVE and DAS-PANEL) to satisfy the requirements of the higher-level scheduler (DAS-ERECT) (figure 3). However, if lower-level scheduling is impossible, the higher-level scheduler attempts to adjust the original requirements. In the constraint-directed graph search, we amalgamated the notions of graph expansion and constraint-directed pruning into an algorithm.

To reduce complexity during the generative scheduling phase and improve the robustness during the reactive scheduling phase, a hierarchical architecture, as shown in figure 3, was adopted. In this architecture, part of the overall scheduling is delegated to the lower-level schedulers—PBS and CBS schedulers—that schedule and control their respective work areas. In our model, each lower-level scheduler can handle its own work area independently as long as it satisfies the requirements imposed by the higher-level dock scheduler.

The characteristics of the hierarchical scheduling architecture can be summarized as follows:

Block Assembly Due-Date Generation by Dock's Generative Scheduler: The dock's *generative scheduler* determines the due date of each block's assembly. In this stage, the total capacities of lower-level assembly shops such as the PBS and the CBS are considered. Then, the scheduler requests the relevant lower-level schedulers to deliver the blocks by these due dates.

Distributed Scheduling and Controlling by PBS and CBS Schedulers: Each scheduler at PBS and CBS schedules its own work area according to the requested due dates as well as the spatial and person-power constraints. When a certain block cannot be assembled in time, the scheduler reports the trouble to the higher-level dock scheduler to obtain assistance in resolving the inconsistency. Then, the due date of a block might be adjusted, or overtime might be required.

Constraint-Directed Graph Search

To find the best erection sequence at the dock, a new search procedure was developed. The first approach we considered was to graph search to expand nodes and select a proper node, possibly using an evaluation function (Nilsson 1980). However, the application of a pure graph search was inappropriate in this case because the measurement of multiple realistic evaluation functions is almost impossible. We also observed that there are precedence constraints between nodes. To accommodate both the graph-expansion framework and the constraint considerations, we proposed an algorithm called the *constraint-directed graph search* (CDGS).

Algorithm of CDGS

To develop an algorithm for the CDGS procedure, we considered the case of a single vessel; however, this algorithm can be extended to multiple-vessel and multiple-dock cases.

Notations N_i: a node *i* (which implies a block in this case), $i = 1, \ldots, m$.

Whole = $\{N_i \mid i = 1, \ldots, m\}$.

KL = potential list of keel-laying blocks, which are the first to be laid for each vessel.

Path = list of scheduled nodes.

Potential = union of adjacent nodes to the ones in current *Path,* excluding those in the *Path* itself.

Erectable = list of nodes that satisfy the technical constraints among the nodes in the current *Potential.*

i-th(*List*) = *i*-th node in the *List* list.

Algorithm

Phase 1. Initialize.

Step 0. Acquire the required data and constraints.

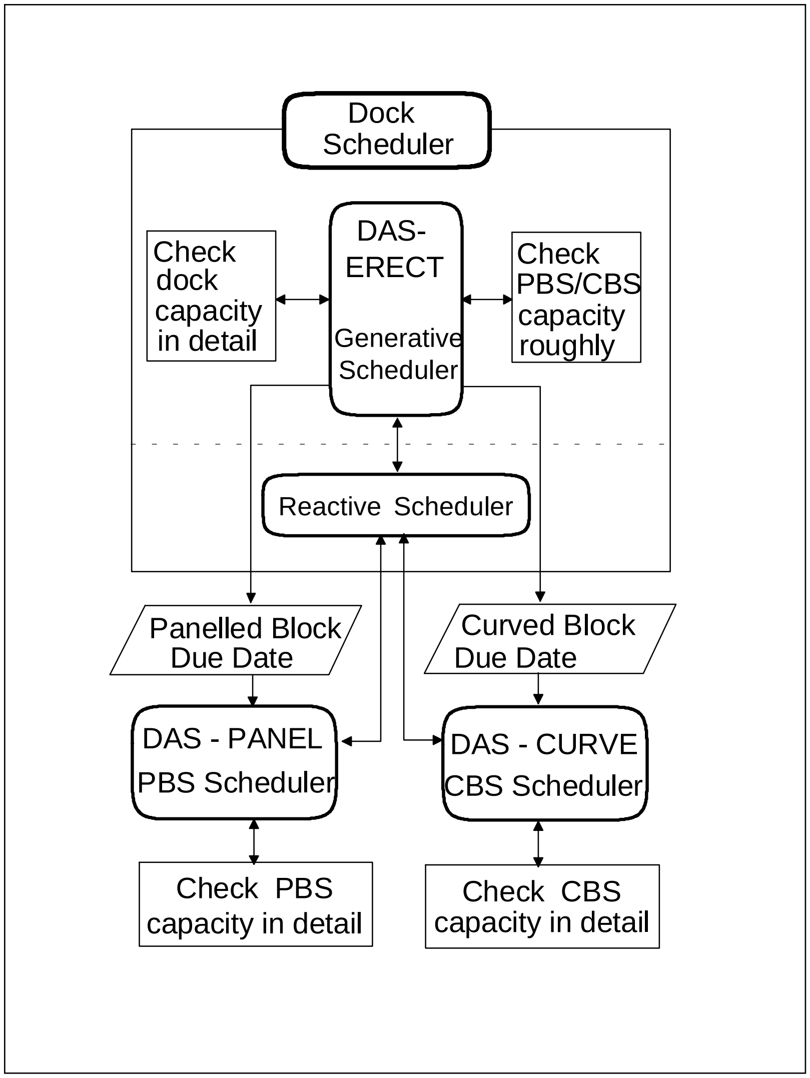


Figure 3. Hierarchical Architecture of Shipbuilding Scheduling.

Step 1. Put all the nodes to be erected in the *Whole list*, and initialize the empty lists.

Whole $\leftarrow \{N_1, \ldots, N_m\}$.

Path $\leftarrow \phi$.

Potential $\leftarrow \phi$.

Erectable $\leftarrow \phi$.

Phase 2. Select keel-laying block.

Step 2. Generate the candidate keel-laying blocks, and store them in *KL*. In this step, the potential keel-laying block-selection constraints are used.

KL $\leftarrow$ Keel_Laying_Constraints (*Whole*).

Step 3. Select a keel-laying block based on the selection strategy. Put the selected keel-laying block in the *Path list.*

$N_k \leftarrow$ Select_a_Keel_Laying_Block (KL).

$KL \leftarrow KL - \{N_k\}$.

$Path \leftarrow Path + \{N_k\}$.

$Whole \leftarrow Whole - Path$.

Phase 3. Expand graph.

Step 4. Expand *Potential* of current *Path* by adding adjacent nodes of N_k while not in the *Path*.

$Potential \leftarrow Potential \cup \{$Adjacent_Node $(N_k) - Path\}$.

Step 5. If the $Potential = \phi$ and $Whole = \phi$, all nodes are erected appropriately. Stop.

If $Potential = \phi$ and $Whole \neq \phi$, the search procedure has problems. Stop with an error message.

Step 6. Among the nodes in *Potential,* select the nodes that also satisfy the technical constraints.

$Erectable \leftarrow$ Technical_Constraints ($Potential$).

Step 7. If $Erectable = \phi$, backtrack.

$N_z \leftarrow last\_th(Path)$.

$Path \leftarrow Path - \{N_z\}$.

$Whole \leftarrow Cons(Whole, N_z)$.

$Potential \leftarrow Potential - \{N_z\} - \{$Added_Potential_by $(N_z)\}$.

$Erectable \leftarrow$ Technical_Constraints ($Potential$).

$N_k \leftarrow$ Evaluate_and_Find_the_Best ($Erectable$).

Then, go to step 4. Otherwise, continue to the next step.

Step 8. Select the best node to be erected from *Erectable*. Update each list.

$N_k \leftarrow$ Evaluate_and_Find_the_Best ($Erectable$).

$Path \leftarrow$ Cons($Path, N_k$).

$Potential \leftarrow Potential - \{N_k\}$.

$Whole \leftarrow Whole - \{N_k\}$.

Go to step 4.

Knowledge Representation for CDGS

The knowledge base used in the CDGS comprises object-oriented data and constraints. Objects represent the hierarchical structure of blocks and superblocks and the resources used in each shop (PBS, CBS, PE shop, dock shop, and so on), as follows:

```
{{ 5060-203
   IS-A: BLOCK
   SHOP: PBS
   PART-OF: MID-BODY
   STRUCTURAL-PART-OF: BOTTOM
 "Relative Position"
   REAR: 5060-232 5060-222
   LEFT: 5060-6A0
   RIGHT: 5060-6B0
   FRONT: 5060-204
   UP: 5060-442 5060-6C0 5060-4B0
   DOWN:
 "Estimated Durations (days) and Man-hours"
   BLOCK-ASSEMBLY-DURATION: 9
   BLOCK-ASSEMBLY-MANHOUR: 346
   DOCK-SETTING-DURATION: 1
   DOCK-SETTING-MANHOUR: 57
   DOCK-FITTING-DURATION: 4
   DOCK-FITTING-MANHOUR: 73
   DOCK-WELDING-DURATION: 3
   DOCK-WELDING-MANHOUR: 73
 "Generated Schedules"
   ASSEMBLY-SST: 92/01/03
   ASSEMBLY-SFT: 92/01/11
   DOCK-SETTING-SST: 92/02/09
   DOCK-SETTING-SFT: 92/02/09
   DOCK-FITTING-SST: 92/02/10
   DOCK-FITTING-SFT: 92/02/13
   DOCK-WELDING-SST: 92/02/11
   DOCK-WELDING-SFT: 92/02/13 }}
```

Note: SST = scheduled start time; SFT = scheduled finish time.

Ten types of technical constraint are acquired and can be grouped into three classes: (1) general technical constraints, (2) constraints on partial sequence, and (3) constraints on precedence relationship. Descriptive examples from each class are as follows:

General technical constraints: "A keel-laying block should be selected among the nonside and bottom blocks of engine room or midbody part."

Constraints on partial sequence: "All the blocks should satisfy the structural stability condition. Therefore, the transverse sequence of the mid-body must satisfy the sequence Bottom → Bulk Head → Side-Shell → Deck."

Constraints on precedence relationship: "The block that needs supporting facilities must be erected after having erected the base blocks on which the facilities should stand."

To illustrate, use three erected blocks as shown in figure 4. Suppose $Path = \{PA, PB, PC\}$. Then, there are 10 potentially erectable blocks. Therefore, $Potential = \{P1, P2, \ldots, P10\}$. The potentially erectable blocks and their positions are shown in figure 5.

We can derive *Erectable* list from the *Potential* list by considering technical constraints such as structural stability. Suppose the *Erectable* list has become {*P*1, *P*2}. To select one block, we evaluate blocks *P*1 and *P*2. Use two criteria: (1) the resource-use level of lower-level assembly shops and (2) the earliest erection start time at the dock. Suppose that blocks *P*1 and *P*2 in the current *Erectable* list are flat bottomed, and they are evaluated by the use level of the PBS. If the block is to be erected on the seventh day at the dock, the block *P*2 should be preferred as the next block to be erected. If *P*2 is selected, the *Path* list becomes {*PA*, *PB*, *PC*, *P*2}. This procedure is repeated until all blocks are erected. Figure 6 shows an erection network scheduled by DAS-ERECT.

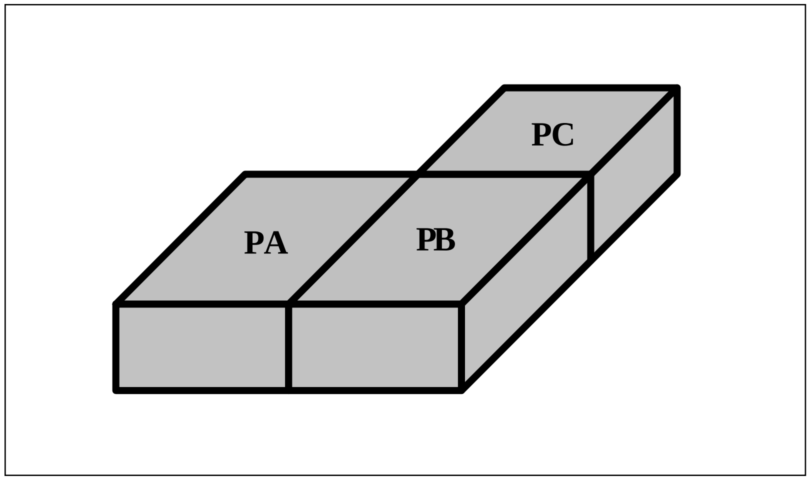


Figure 4. Partially Erected Blocks.

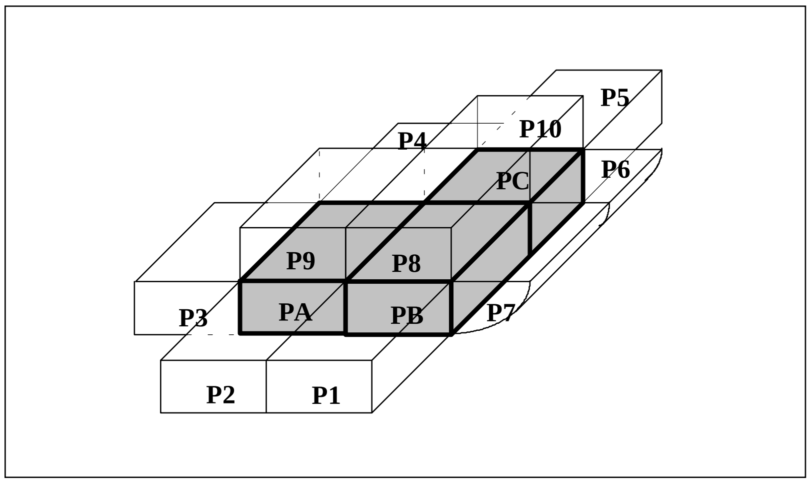


Figure 5. Spatial Position of Potentially Erectable Blocks.

Spatial Scheduling in Shipbuilding

In the shipbuilding plants that handle the heavy and bulky blocks, it is necessary to use expensive material-handling equipment such as cranes and work plates. Because the space equipped with such facilities is typically limited and bottlenecked, the scheduling needs to consider the spatial resources as well as traditional resources such as personnel and machines. We call this kind of scheduling *spatial scheduling*. As the term implies, spatial scheduling deals with the optimal dynamic spatial layout schedules. In a shipyard, spatial scheduling problems occur frequently in various working areas such as erection docks, preerection shops, and the block-assembly shop.

To date, human schedulers have carried out the spatial scheduling manually, using no automated aids. Even though human experts have extensive experience in spatial scheduling, it takes a long time and much effort to produce a satisfactory schedule because of the huge search space required to consider the blocks' geometric shapes. For example, spatial scheduling for six months is beyond the scope of a human's mental capacity; so, it was impossible to build such a large-scale spatial schedule in advance. Nevertheless, the spatial resource tends to be a critical resource in shipbuilding. Therefore, automation of the spatial scheduling process was critical to improve productivity in the shipbuilding plants and achieve the total integration of scheduling expert systems.

At Daewoo, some attempts have been made to solve the spatial scheduling problem. One approach was a simple spatial scheduling system that assumed that the shape of blocks can be approximated to a rectangle. However, the field schedulers rejected it because the approximation sacrificed spatial use too much. DAS-CURVE assumes the approximation of the blocks' shape to a polygon, on which the schedulers agree.

Another approach that failed was the interactive spatial scheduling support system that provided the graphic user interface for the human scheduler. The interactive system was not effective because it couldn't automate the tedious spatial scheduling process and reduce the information burden and scheduling time. It could only replace the paper and pencil with a computerized interface.

The objectives of spatial scheduling can vary somewhat depending on the nature of a given plant. In general, however, spatial scheduling systems pursue the objectives of due-date satisfaction, maximal use of spatial and nonspatial resources, and minimization of waiting times for work-in-process and final-product inventories.

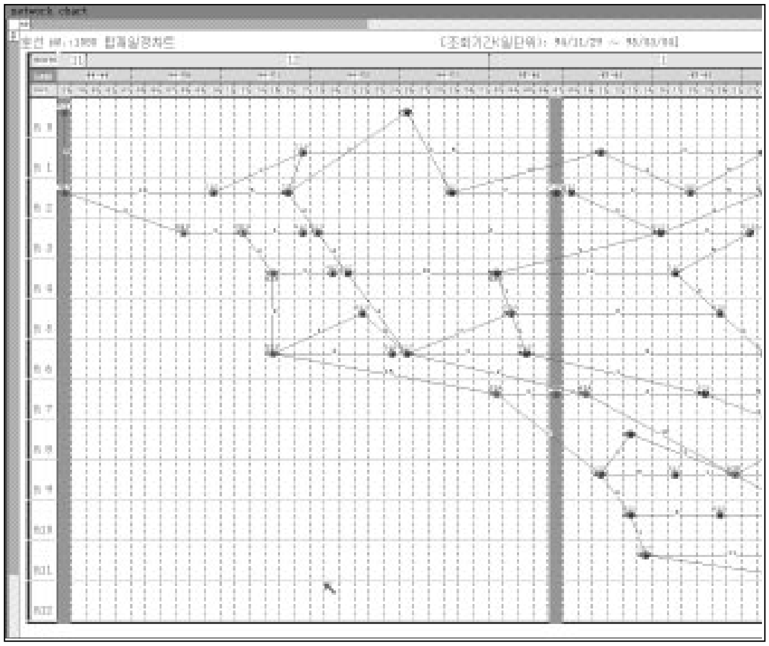


Figure 6. An Erection Network Scheduled by DAS-ERECT.

Typical constraints include crane capacity, person-hour availability, assembly due date, precedence between associated assemblies, physical adjacency of coupled objects for operational efficiency, minimum required distance between blocks, and maximum acceptable waiting times for completed and work-in-process blocks.

Typical necessary input data include jobs with due dates and their constituent activities, two-dimensional geometric spatial objects of the activities, required processing times for each activity, and spatial shapes of work plates.

In the shipbuilding domain, the shapes of most objects tend to be convex polygons such as triangles, rectangles, or trapezoids. Some blocks might have some local concavity. However, in most cases, the local concave space is not usable by other objects. Therefore, these objects can be approximated as convex polygons.

In shipbuilding, the orientation of an object is prefixed to four alternative orientations (0^o, 90^o, 180^o, and 270^o) to ensure stable crane operations.

Search Space in Spatial Scheduling

To find the feasible locatable positions of an object a_i within a workplate W that do not overlap a scheduled object b_j, we can adopt the notion of configuration space (Zhu and Latombe 1991; Lozano-Perez 1983). *Configuration space* is the space through which the reference point of an object (a robot, for example) with fixed orientation can possibly pass without colliding with obstacles that are present. In our research, there are two kinds of configuration space: (1) the obstacle-avoiding space and (2) the inner locatable space. In the *obstacle-avoiding space* $S(a_i|B)$, the reference point of an object a_i can be located without colliding with the already located objects that are regarded as obstacles. In the *inner locatable space* $S(a_i|W)$, the reference point of an object a_i can be located within the boundary of a work area W. Thus, the *feasible locatable space* $S(a_i|B,W)$ can be derived by intersecting the previous two spaces: $S(a_i|B,W) = S(a_i|B) \cap S(a_i|W)$.

Figure 7 illustrates the spaces where the object a_1 is to be located within W, on which two objects b_1 and b_2 are already located. Because the feasible locatable space is a continuous space, it is impossible to find all the points in it. To extract a set of meaningful discrete points out of the continuous space, we define the *distinctive locatable point set* $D(a_i|B,W)$, which consists of the vertexes of the feasible locatable space.

Theoretically speaking, the distinctive locatable point set does not guarantee finding an optimal location (Lee and Lee 1995). However, the points have empirically provided satisfactory locations with the advantage of computational efficiency.

The distinctive locatable points can be classified into four categories: (1) $D_O(a_i|B,W)$, the union of the feasible vertexes of each obstacle-avoiding space; (2) $D_I(a_i|B,W)$, the feasible vertexes of the inner locatable space; (3) $D_{OI}(a_i|B,W)$, the union of the feasible intersection points between the boundaries of each obstacle-avoiding space and the inner locatable space; and (4) $D_{OO}(a_i|B,W)$, the union of the feasible intersection points between the boundaries of each obstacle-avoiding space.

The following points are illustrated in figure 7: three $D_O(a_i|B,W)$, marked as triangles; three $D_I(a_i|B,W)$, marked as squares; six $D_{OI}(a_i|B,W)$, marked as circles; and two $D_{OO}(a_i|B,W)$, marked as diamonds. The categorized points can be used to reduce the search space by identifying a special category of the distinctive locatable point set contingent to the situation.

Search in the Distinctive Locatable Point Set

Because most layout problems are NP complete, we propose four positioning strategies that can effectively be applied contingent to the situation: (1) the maximal remnant space use strategy, (2) the maximal free rectangular space strategy, (3) the initial positioning strategy, and (4) the edging strategy.

Because each strategy has its own merits depending on the situation, we synthesized a composite positioning algorithm that can apply an adequate positioning strategy contingently. Key issues in composite positioning are identification of a situation, reduction of search space, and selection of effective strategy. In this study, we identified four types of situation that depend on the existence of already located objects, the attempted location (near corner, edge, or other object), and the shapes of the objects. The strategies are by no means complete. However, we empirically verified that these four situation types can still effectively capture the possible situations.

Composite Positioning Algorithm

Compute $D(a_i|B,W) = [d_1,d_2, \ldots , d_k, \ldots , d_z]$.
CS: = $D(a_i|B,W)$ /* CS = a set of candidate solutions */
If CS = Null
  Then /* a_i cannot be allocated in W without changing the current schedule */
  Start backtracking.
Else
  If there are no located objects yet
    Then reset $CS := D_I(a_i|W)$.
    If a_i is not rectangular
      Then apply the initial positioning strategy.
    Else /* a_i is rectangular */
      Then select a point arbitrarily.
  Else /* there are no located objects yet*/
    If most b_js are rectangular or near rectangular
      Then apply the maximal free rectangular space strategy.
    Else /* most b_js are not rectangular */
      If $CS = D_{OI}(a_i|B,W)$
        Then apply the edging strategy.
      Else
        Then apply the maximal remnant space-use strategy.

Backtracking and Adjustment in Spatial Scheduling

If we cannot find a feasible schedule for a certain day, we have to backtrack to adjust the current spatial layout, the starting times of already scheduled activities, and the resource commitment level. Some of the backtracking adjustment could have been avoided if we could have looked ahead to needs over the next several days. However, obtaining information on the precise impact of these future needs is almost as expensive as the scheduling itself. Therefore, we adopted the backtracking and adjustment strategy.

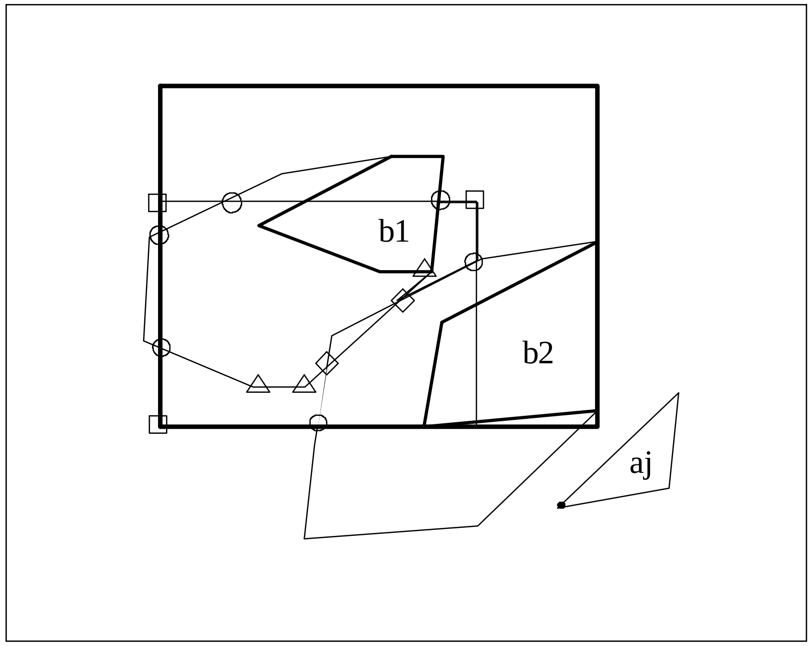


Figure 7. Feasible Locatable Space and Distinctive Locatable Points.

For the shipbuilding domain, we used the following six types of adjustment: (1) work-plate reselection, (2) intraplate spatial adjustment, (3) interplate spatial adjustment, (4) intraplate temporal adjustment, (5) intraplate spatiotemporal adjustment, and (6) interplate spatiotemporal adjustment.

Principles for the spatial scheduler to follow in sequencing the application of these adjustment types can be summarized as follows:

First, monitor the status of the free rectangular space of each work plate as a measure of spaciousness. For a new object, assign a work plate according to this criterion as long as the special constraints for the new object are satisfied. However, if the shape of the free space turns out to be unsuitable for accommodating the particular shape of a new object, then reselect the work plate if other sufficiently spacious options exist.

Second, apply the spatial adjustment (including the work-plate reselection) before applying the temporal adjustment because the temporal adjustment invokes a trade-off between the tardiness of new activity and the work-in-process inventory of allocated activities. However, spatial adjustment does not incur such a cost as long as the constraint of operational adjacency among certain blocks is not violated.

Third, apply the intraplate adjustment before the interplate adjustment because the intraplate search is intuitively a more efficient option.

Fourth, apply the purely spatial or temporal adjustment strategy first before the spatiotemporal adjustment because the composite spatiotemporal adjustment requires greater computation than the single-adjustment strategy.

The sequencing rules in applying the adjustment strategies do not have to be sequential. However, the computational effort to identify the optimal-adjustment strategy is expensive, as it is with the lookahead strategy. Therefore, it is desirable to have a reasonable predetermined default sequence. In this study, the priority of adjustment strategies was incorporated into the adjustment algorithm based on this rationale.

Another default scheduling strategy adopted here with respect to the starting time of activities was the *latest starting time strategy*: "Assign the activity to be finished one day before the due date to avoid the wait of work-in-process inventory. However, if this time schedule is not possible to achieve, try to minimize the waiting time."

The contingent choice of the proper adjustment strategy is related to the cause of scheduling failure and the scheduling status. For example,

If the cause of the scheduling failure is deficiency of manpower
Then skip the intraplate spatial adjustment because it will not end successfully.

If the space use of a work plate is relatively low
Then the intraplate spatial adjustment is more effective than the intraplate temporal adjustment.

If the space use of a work plate is relatively high
Then the intraplate temporal adjustment is more effective than the intraplate spatial adjustment.

If the space use of every work plate is very high
Then skip the interplate spatial adjustment.

DAS-CURVE: Curved-Block Assembly Shop Scheduler

DAS-CURVE is a representative spatial scheduling expert system (Lee, Lee, and Choi 1995; Lee and Lee 1992) for the curved-bottom block assembly shop, first applying UNIK (unified knowledge)-SPACE. The system generates the spatial schedule of assembling blocks to meet the due dates imposed by DAS-ERECT. The block assembly shop has about 15 work plates, with cranes to lift blocks and subassemblies. The resources are limited by the availability of the spatial work plates as well as the nonspatial resources of personnel and cranes. Figure 8 illustrates an output screen of DAS-CURVE that shows a snapshot of the spatial layout status of the eight work plates in the shop for one day. Figure 9 illustrates the dynamic spatial layouts of a work plate during an indicated time interval.

As the term *spatial scheduling* implies, visual interactive scheduling is an essential feature for the scheduler's initiative. Therefore, DAS-CURVE is equipped with a mouse-based graphic user interface. To maintain consistency between the user's visual interactive input and the invisible constraints, the reactive scheduling capability should work behind the screen, typically adopting the adjustment methods described earlier.

Spatial Scheduling in Other Systems

In erection scheduling, the spatial scheduling problem occurs in the preerection shop near the docks. The preerection shop shares many similarities with the curved-bottom block assembly shop, but it also has some special spatial layout constraints. These special constraints result from the differences in the technical constraints between the two shops. The spatial scheduling module reflecting these features is imbedded in DAS-ERECT.

The long-range planning of a shipyard should consider the spatial capacities of docks because the sizes and the shapes of ships are diverse. Especially when there are hundreds of product-mix candidates, automated checking of spatial availability is essential because of the various spatial constraints on the layout of ships that exist because of the technical constraints during the erection process. DAS-LPP has the dynamic spatial layout module using UNIK-SPACE (Lee, Lee, and Choi 1994).

To implement the spatial scheduling methodology, many computational geometric algorithms are used, such as the *point-in-polygon algorithm* (Preparata and Shamos 1985) to determine whether a point is in a

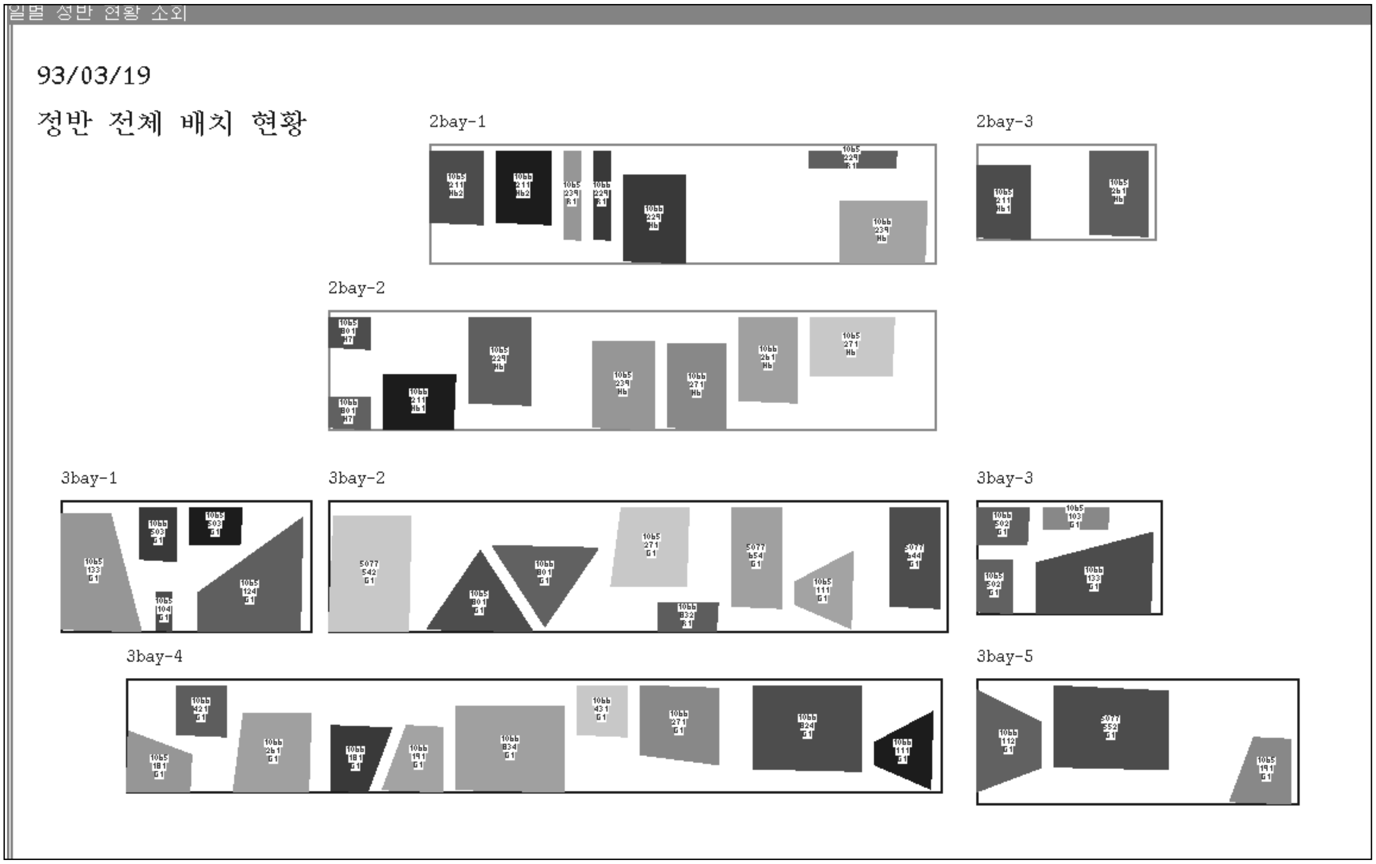


Figure 8. An Output Screen Showing a Snapshot of Spatial Layout in DAS-CURVE.

convex polygon, the *polygon-intersection algorithm* (O'Rourke et al. 1982) to compute the intersection of two convex polygons, and the *polygon setsum algorithm* (Lozano-Perez 1983) to compute the setsum of two convex polygons. In addition to these known algorithms, various specialized computational geometric algorithms are devised for spatial scheduling, such as those for enlarging polygons to the extent of the safety distance and computing feasible intersection points among polygons.

DAS-PANEL: Paneled-Block Assembly Shop Scheduler

DAS-PANEL (Hong, Kim, and Lee 1993) is a scheduling expert system for the paneled-block assembly shop in which blocks and their subassemblies are welded on the assembly lines and adjacent off lines, respectively. Because the block size and compositions of subassemblies are diverse, the scheduling should encompass the detailed scheduling of subassemblies as well as the main assembly line scheduling of blocks. DAS-PANEL has two characteristics: (1) Because the blocks made in the shop are one of a kind, the speed of the assembly line should not be fixed in advance. To improve the productivity of the assembly line, it is necessary to generate a schedule that can dynamically change the cycle time according to the characteristics of the blocks. (2) Its objective is to minimize not only the assembly time of blocks on the main line but also the waiting times of subassemblies and blocks. DAS-PANEL was built using a typical forward-chaining rule-based tool, UNIK-FWD, written in C++. Figure 10 shows the dynamically changing cycle time generated by DAS-PANEL.

DAS-MH: Neural Network–Based Man-Hour Estimator

To establish reliable scheduling expert systems, the estimation of accurate person-hour requirements for each assembly is a prerequisite. To supply the required welding person-hour input of each one-of-a-kind block for DAS scheduling systems, we adopted the artificial neural network as an estimator (Lee and Kim 1994).

To build a reliable and efficient neural network model, we followed this research proce-

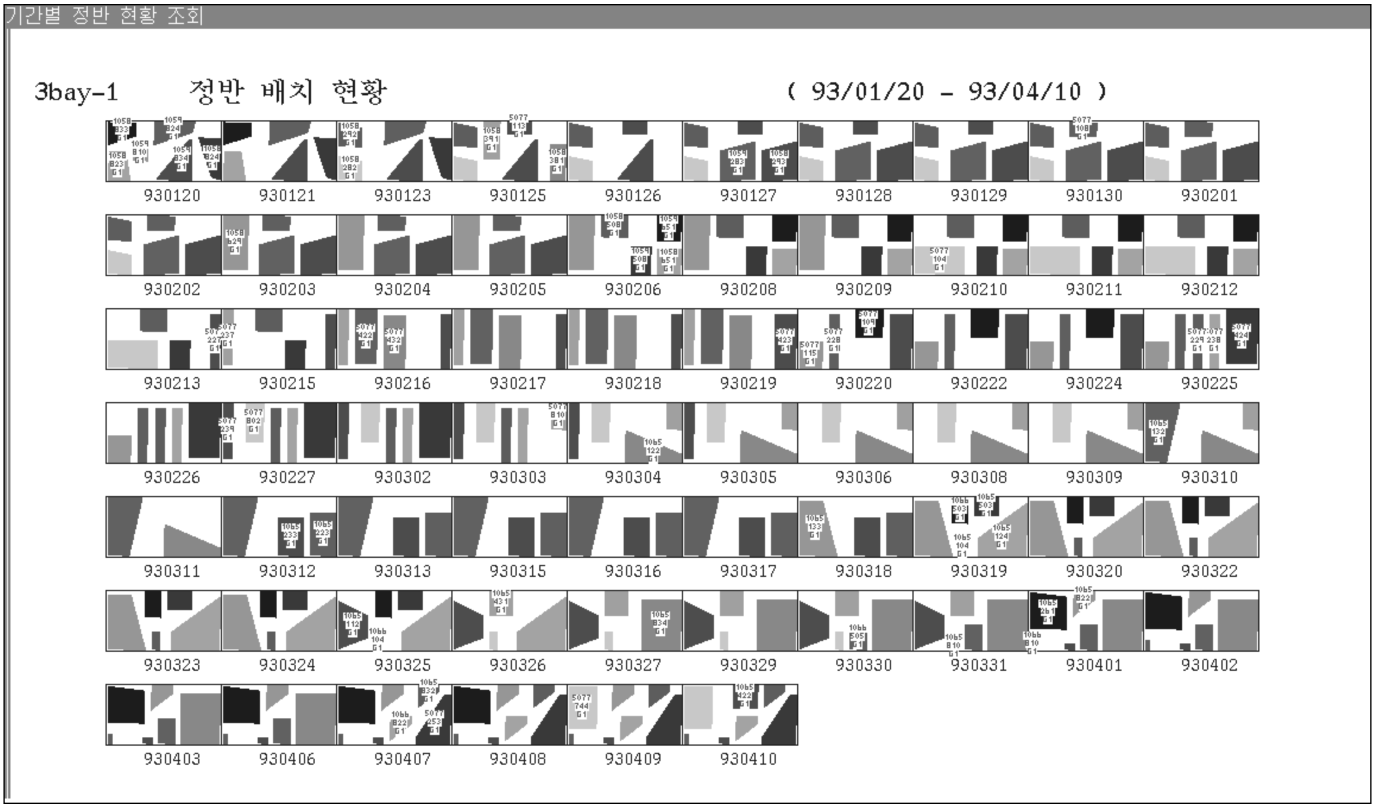


Figure 9. An Output Screen Showing a Dynamic Spatial Layout of a Work Plate in DAS-CURVE.

dure: (1) Select candidate input variables: Four categories of variables are selected with possible values for each variable. (2) Eliminate unnecessary variables: Highly correlated redundant cardinal variables are filtered out by stepwise regression as a preprocessor. (3) Train and test the network with and without the preprocessing. (4) Compare the estimation performance by the neural network with one by the regression analysis.

We selected four categories of variable that influence the welding time. They are candidate input variables of the neural network model.

First is ship type. For use, select one of the following values: VLCC (very large crude oil carrier), OBO (ore bulk carrier), COT (crude oil tanker), B/C (bulk carrier), CONT (container), P/C (product carrier), and so on. The *dead weight*, the weight that the ship can carry, must also be considered.

Second is block type. For locus, select one of the following values: side-shell, bulbus-low, engine-room-side-upper, fore-deck, longitudinal-bulk-head, transverse-bulk-head, transverse-noncorrugated-bulk-head, mid-ship-bottom, and so on. Whether it is a paneled or a curved bottom or a single or a double shell must also be considered.

Third is a block's physical characteristics: welding length, joint length, and block's weight.

Fourth is shop type. For the assembly shop, select one of the following values: 3DS (three-dimensional shop), BOS (building outfitting shop), PBS (paneled-block shop), and A-7 (area code no. 7). Whether it is indoor or outdoor must be considered for each shop. In total, we italicized 10 candidate input variables: 4 cardinal- and 6 nominal-scale variables.

To eliminate unnecessary variables, we adopted the stepwise regression approach with four cardinal variables: (1) dead weight, (2) welding length, (3) joint length, and (4) block weight.

We expected that multicollinear variables had to be dropped to keep the model robust. In this manner, the joint length and the block weight are finally selected. Thus, eight variables are eventually selected as the input to the neural network. Figure 11 shows a configuration of the neural network developed using the neural network developer UNIK-NEURO.

To confirm whether the neural network model significantly outperforms other possible approaches, we compared it with two approaches: (1) multiple regression and (2) simple regression through origin. We made the comparison using simple regression

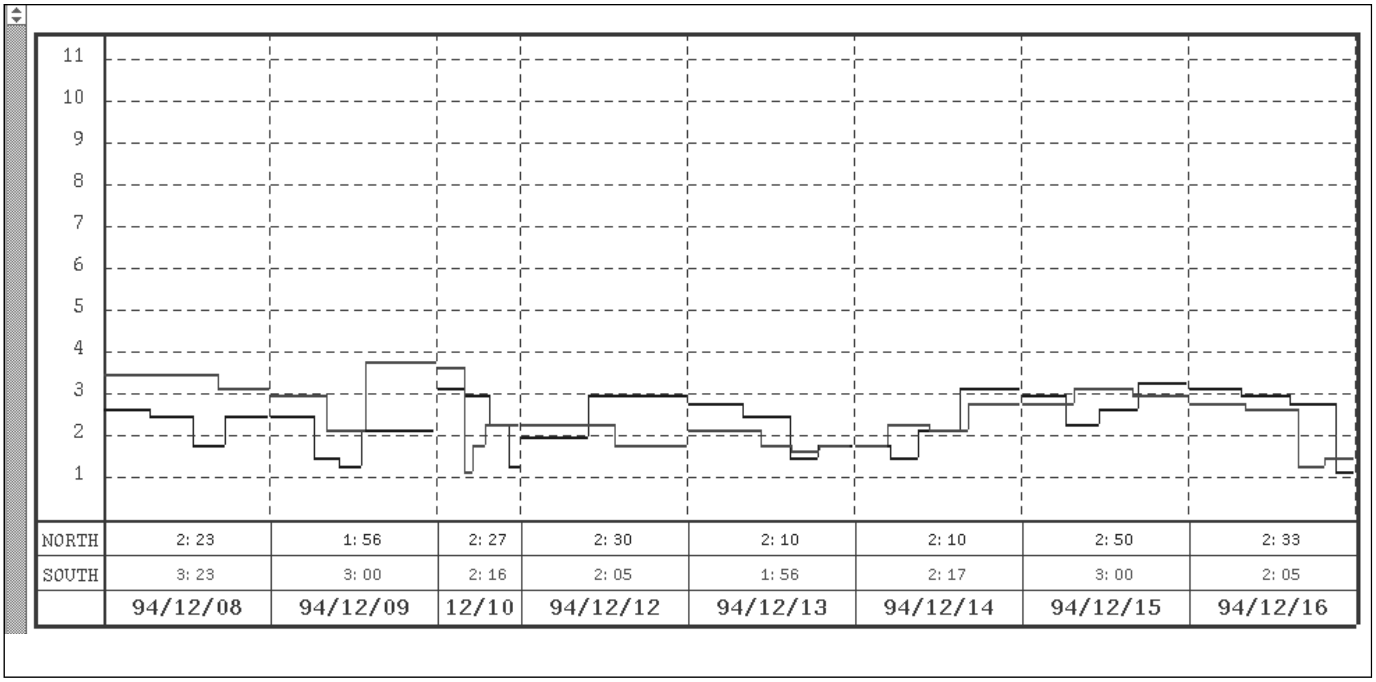


Figure 10. An Output Screen Showing the Dynamically Changing Cycle Time in DAS-PANEL.

through origin because this method has been used in the field for years with the name *unit estimator*. To estimate using the unit-estimation method, the joint length of the block is used as an independent variable for each group of blocks, which are grouped into 17 families. The neural network model significantly outperformed the multiple-regression and the simple-regression-through-origin approaches.

DAS-LPP: Long-Range Production Planning Expert System

The system aims to automate the generation of the shipbuilding production program, which is the starting point of production management and the important criteria for marketing activities (Yang and Choi 1994). To balance loads of production processes keeping the due dates of ship deliveries, we developed the expert system using the beam-search module, which is connected with the linear programming solver MINOS. We used linear programming to model the cumulative load curve (so-called S-curve, as in figure 12) and the beam-search procedure to select the best product-mix cases that reflect the shipbuilding plant's status.

Implementation

For this project, we used the frame-based expert system tool UNIK, which was developed by KAIST, upgraded from the initial Lisp version to a C version. Because we own the source codes and can improve them, it was an adequate tool for this kind of project, which requires flexible adjustment. UNIK has features such as forward chaining (UNIK-FWD), backward chaining (UNIK-BWD), inductive learning (UNIK-INDUCE), rule generation from diagrammatic representation (UNIK-RULEGEN), and neural network learning (UNIK-NEURO) as well as a library of Lisplike functions (UNIK-KERNEL). UNIK has versions operating on UNIX, DOS, and WINDOWS. DAS systems were implemented on the Sun SPARCSTATION 10 series.

Development Strategy and Deployment Process

For the large-scale and long-range project, a three-stage development process was devised (Lee 1993), as summarized in table 1. In this phased approach, we considered not only user requirement but also data availability. The first stage was vision revelation. The KAIST team professed that the DAS Project was not mere system development but theoretical research on the following issues: (1) constraint-directed graph search for erection scheduling, considering the work loads at the indoor shops; (2) spatial scheduling; (3) line balancing with flexible process planning and dynamic cycle time; (4) processing time estimation using neural networks; and (5) inter-

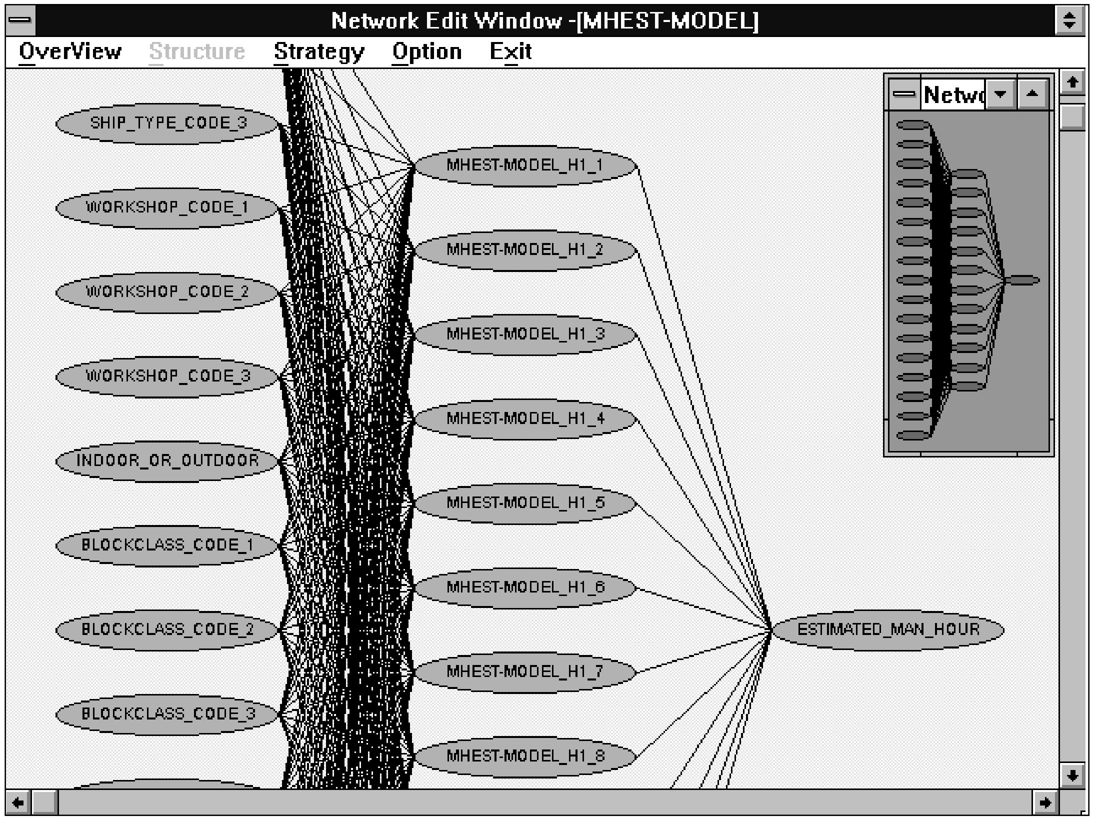


Figure 11. A Neural Network Configuration for Person-Hour Estimation.

face and coordination among multiple expert systems and databases.

These issues were explored during the project's first year, 1991, and prototypes of the DAS systems were developed using UNIK. These prototypes could successfully demonstrate the vision, although they could not provide the satisfactory speed.

The second stage was data-dependent realization. Daewoo employees were frustrated because they couldn't provide the data necessary to run the prototype systems. However, it was a good point at which to identify which data were available and which were not. This process required close communication with the design division because the initial prototype required information about the design specification from the computer-aided design (CAD) tools to automate the process planning and person-hour estimation. However, these data could not be provided until the current CAD system was upgraded to a solid modeler, which was not technically feasible at the time. Other data that could not be supported were the erection sequence–dependent person-hour and the processing time requirements. Therefore, based on the data availability, the prototype systems were degraded. However, management was delighted because the systems could run based on the currently available data, although they were not ideal. The systems in the second stage were developed using a C++ version of UNIK to enhance operational speeds. The systems were installed in the field in 1992 and tested under real-world conditions.

The third stage was prospective enhancement. During this stage, the systems developed in the second stage were incrementally enhanced. A difference is that the enhancement was oriented toward the target set at the end of the first stage. The systems gradually improved because more data became available during 1993. This enhancement will continue as necessary data are collected.

Project Work Group Organization

The project team consisted of both Daewoo and KAIST personnel (table 2). We observed that the roles of the project members evolved according to the development phases, as shown in table 2. Working and communicat-

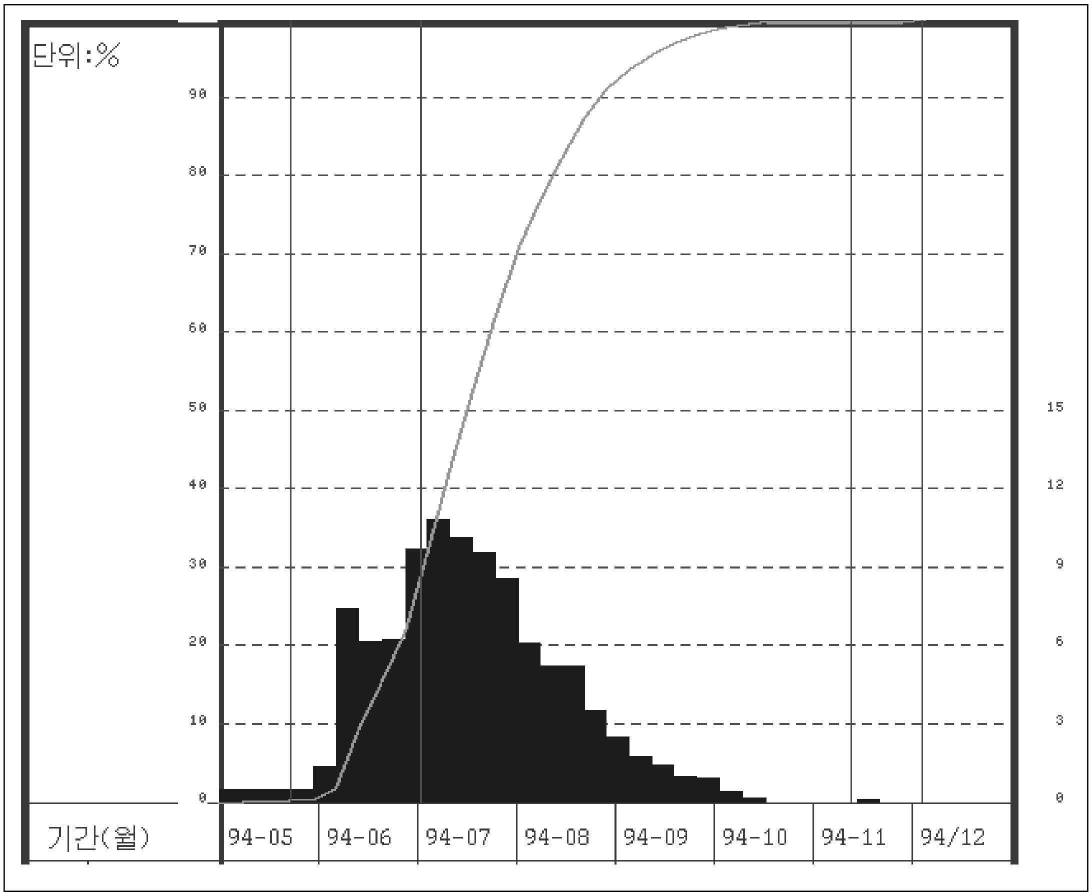


Figure 12. A Cumulative Load Curve (S-curve) Used in DAS-LPP.

ing with management was smooth because the executives involved in the project did not change until the end of the project.

Innovations at the Site

The innovations made possible by using the DAS systems can be summarized as follows:

Reengineering of the planning and working processes: The aim of DAS was not automating the existing processes but innovating the old processes using AI technology. We present three examples of these innovations (Kim 1994). First is the product-mix case-selection **process.** Before using DAS-LPP, the loads of the plants were calculated after product-mix cases were selected. Now, we can select good product-mix cases considering the future loads of the plants. Second is the spatial scheduling process. Before using the spatial scheduling expert systems such as DAS-CURVE, the spatial allocation plan was made after temporal allocation planning. Now, the integration of two scheduling processes saves much time and effort and enhances the quality of the schedule. Third is the expected person-hour calculation process. The previous inaccurate and heuristic unit method was replaced by the neural network–based estimation method.

Selection of the best schedule from simulation: The speedup of the schedule-generation time enabled the field scheduler to try multiple scheduling strategies and select the best schedule among the simulation results.

Move of the scheduling effort from simultaneous and centered scheduling: The networked scheduling expert systems in a hierarchical architecture enabled the production management department to plan and control the plants, reducing the negotiation efforts among plants.

Visual production management and control by graphic output: The graphic output screens generated from systems such as DAS-CURVE improved the planning and work productivity using clear communication between management and workers.

Technology transfer from university to industry: The DAS Project is famous in Korea as a typical success story of the cooperation between university and industry. Now Dae-

	First Year, 1991	Second Year, 1992	Third Year, 1993
DAS-ERECT	Research on constraint-directed graph search	Spatial scheduling in preerection shop, system implementation, graphic user interface implementation	Experiments, graphic user interface development, system upgrade and maintenance
DAS-CURVE	Research on spatial scheduling	System implementation, graphic user interface implementation	
DAS-PANEL	Research on line balancing with dynamic cycle time	Rule base and inference engine construction, system and graphic user interface implementation	
DAS-MH	Tool (UNIK-NEURO) development	Neural network modeling and system development	Experiment and model selection
DAS-LPP		Research on CSP and LP modeling of S-curve	Interactive planning prototype
Implementation Language and Tool	Lisp version prototype	Operational system (C conversion)	Installed in field
Development Phase	Vision revelation	Data-dependent realization	Prospective enhancement

Table 1. Development Process.

woo has the ability to develop expert systems by itself.

Development Cost

The development costs for the three years were calculated at approximately $159,000 for the hardware and $675,000 for the research and one developer from each organization. The total cost was about $834,000.

Estimate of Payoff

Although the revenue implication for the project is not easy to calculate, the company estimates were based on the expected contributing rates of the production productivity and planning productivity improvements. Because the estimated contribution rate of DAS to the yearly production productivity improvement (15 percent) is 30 percent and to the planning productivity improvement is 50 percent, the expected annual benefit of the DAS Project is about $4 million.

Maintenance

The systems are maintained by the programmers and knowledge engineers who were trained during the project. The schedulers can set the contingent scheduling strategy by establishing the parameters for the constraint base and the rule base. For flexible system maintenance based on the change of plant layout and facilities, KAIST and Daewoo are currently conducting research on the scheduling system generator as part of the DAS-II Project.

Acknowledgments

This research is funded by Daewoo Heavy Industries Ltd. in Korea. We are particularly grateful to Vice-President Dong Kyu Park, Executive Directors Wan Chul Suh, Yang Bin Cho, and Hee Gyu Kim, Department Manager Se Jin Jang, Section Manager Gwang Joo Lee, and Sang Soon Im and Yong Chul Kim for their cooperation during knowledge acquisition. We also would like to thank members of the DAS Project team at KAIST and Daewoo: Jae Wook Kim, Jeong Seung Lee, Jae Ho Chun, and Seok Jun Byun.

	First Year, 1991	Second Year, 1992	Third Year, 1993	Roles
University (KAIST)	9 persons	9 persons	6 persons	Research and development, technology transfer
	Research and prototype development, knowledge engineering	Inference engine development, tool and system conversion (Lisp to C)	Advanced implementation and experiment	
People from company	3 persons	3 persons	5 persons	3-year residence at the university
	Specification, transfer, learning, and training	Rule base and graphic user interface development	Maintenance and graphic user interface development	
Management team in the company		4 persons	4 persons	Coordination, on-the-job training
		Project Management	Data Support and Training	
Domain experts		11 persons	17 persons	In-house training, practice, and testing, support of scheduling masters
		Knowledge transfer and feedback, prototype testing	Use learning and system testing	

Table 2. The Number of Members in Each Group and Their Changing Roles.

References

Hong, J. S.; Kim, E. Y.; and Lee, J. K. 1993. Assembly Line Scheduling Expert Systems in Shipbuilding: DAS-PANEL. In Proceedings of the Korean Expert Systems Society Conference (in Korean), 147–158. Seoul, Korea: Korean Expert Systems Society.

Kim, E. Y. 1994. CIM in Shipbuilding: A Case Study. In Proceedings of the Korean Expert Systems Society Conference (in Korean), 81–98. Seoul, Korea: Korean Expert Systems Society.

Lee, J. K. 1993. Phased Development Strategy for Complex Expert Systems: A Shipbuilding Scheduling Case. In Proceedings of 93' Pan Pacific Conference on Information Systems, 255–257. Taiwan, Republic of China: PACIS.

Lee, J. K., and Kim, H. D. 1994. Man-Hour Requirement Estimation for Assemblies Using Neural Networks. In Proceedings of '94 Japan/Korea Joint Conference on Expert Systems, 203–206. Tokyo, Japan: Japanese Society for Artificial Intelligence.

Lee, K. J., and Lee, J. K. 1995. The Dominance and Preference of Search Space in Dynamic Spatial Layout, Working Paper, KAIST-MIS-WP-9503-01, Department of Management Information Systems, Korean Advanced Institute of Science and Technology.

Lee, K. J., and Lee, J. K. 1992. Spatial Scheduling and Its Application to Shipbuilding. In Proceedings of the Second Pacific Rim Conference on Artificial Intelligence, 1183–1189. Seoul, Korea: Korean Information Science Society.

Lee, K. J.; Lee, J. K.; and Choi, S. Y. 1995. A Spatial Scheduling System and Its Application to Shipbuilding: DAS-CURVE. *Robotics and Computer Integrated Manufacturing*. Forthcoming.

Lee, K. J.; Lee, J. K.; and Choi, S. Y. 1994. Spatial Scheduling Expert Systems for Shipbuilding. In Proceedings of the Second World Congress on Expert Systems, 243–249. New York: Cognizant Communication Corporation.

Lee, J. K.; Suh, M. S.; and Fox, M. S. 1993. Contingencies for the Design of Scheduling Expert Systems. *Expert Systems with Applications* 6:219–230.

Lee, J. K.; Choi, H. R.; Yang, O. R.; and Kim, H. D. 1994. Scheduling Shipbuilding Using a Constraint-Directed Graph Search: DAS-ERECT. *Intelligent Systems in Accounting, Finance, and Management* 3:111–125.

Lozano-Pérez, T. 1983. Spatial Planning: A Configuration Space Approach. *IEEE Transactions on Computers* 32(2): 108–120.

Nilsson, N. J. 1980. *Principles of Artificial Intelligence*. San Francisco, Calif.: Morgan Kaufmann.

O'Rourke, J.; Chien, C.; Olson, T.; and Naddor, D. 1982. A New Linear Algorithm for Intersecting Convex Polygons. *Computer Graphics and Image Processing* 19:384–391.

Preparata, F. P., and Shamos, M. 1985. *Computational Geometry: An Introduction*. Reading, N.Y.: Springer Verlag.

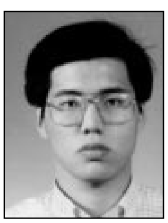
Yang, O. R., and Choi, S. Y. 1994. An Expert System with Optimization Model: DAS-LPP. In Proceedings of Korean Expert Systems Society Conference (in Korean), 241–251. Seoul, Korea: Korean Expert Systems Society.

Zhu, D., and Latombe, J. C. 1991. Mechanization of

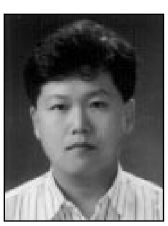
Spatial Reasoning for Automatic Pipe Layout Design. *AI EDAM* 5(1): 1–20.

Jae Kyu Lee is a professor of management information systems at Korea Advanced Institute of Science and Technology. He received a B.A. from Seoul National University; an M.S. from the Korea Advanced Institute of Science and Technology; and a Ph.D. from the Wharton School, University of Pennsylvania. He has authored several books on expert systems and published numerous papers. Currently, he is on the editorial boards of the journals *Decision Support Systems; Expert Systems with Applications: An International Journal; International Journal of Intelligent Systems in Accounting, Finance, and Management;* and *New Review of Applied Expert Systems*. He is a member of the American Association for Artificial Intelligence.

Kyoung Jun Lee is a senior researcher in the Department of Management Information Systems at the Korea Advanced Institute of Science and Technology (KAIST). He received his B.S. and M.S. in management science from KAIST. His Ph.D. in industrial management from KAIST is in the field of AI, where his dissertation dealt with the development of spatial scheduling expert systems. His major research interests include spatiotemporal reasoning, intelligent spatial information systems, AI for electronic commerce, and intelligent manufacturing systems. He is a member of the American Association for Artificial Intelligence.

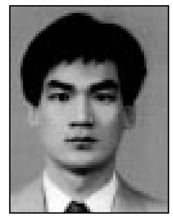


June Seok Hong received a bachelor's degree in business administration from Seoul National University in 1989 and an M.S. in management science from KAIST in 1991. He is a Ph.D. candidate in management science at KAIST. His research interests include expert system scheduling, knowledge engineering, and constraint-satisfaction problems.



Wooju Kim is a senior researcher in the Department of Management Information Systems at the Korea Advanced Institute of Science and Technology (KAIST). He received his B.B.A. from Yonsei University in 1987 and his M.Sc. from KAIST in 1989. He also received his Ph.D. in management science from KAIST in 1994. His major research interests include neural networks, expert systems, software engineering, knowledge representation and acquisition, and managerial forecasting. He is a member of the Korea Expert Systems Society.

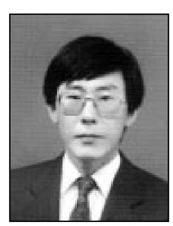
Eun Young Kim is a deputy general manager at Daewoo Information Systems Co., Ltd. He received his bachelor's degree in mechanical engineering from Ajou University. His major interests include expert systems, computer-integrated manufacturing, and electronic commerce. He is a member of the Korea Expert Systems Society.



Soo Yeoul Choi is a section manager at Daewoo Information Systems Co., Ltd. He received his bachelor's degree in industrial engineering from Inha University. His major interests include expert systems and computer-integrated manufacturing. He is a member of the Korea Expert Systems Society.

Ho Dong Kim is a section manager in the Department of Consulting Business at Daewoo Information Systems Co., Ltd. He received his bachelor's degree in naval architecture from Seoul National University and holds an M.S. in management information systems from the Korea Advanced Institute of Science and Technology. His major research interests include the application of neural networks and computer-integrated manufacturing.

Ok Ryul Yang is an assistant section manager at Daewoo Information Systems Co., Ltd. He holds a B.B.A. from Seoul National University and an M.S. from the Korea Advanced Institute of Science and Technology. His research interests include intelligent scheduling systems, wireless data communication, and multimedia and traffic information systems.



Hyung Rim Choi is a professor of management information systems at the Dong-A University. He received his Ph.D. in management science from the Korea Advanced Institute of Science and Technology. His major research focus is on the automation of process planning and scheduling in manufacturing domains. Current research interests include applications of various AI technologies, such as case-based reasoning and genetic algorithms, to shipbuilding.