A comparative study of analytical tools for strategic and tactical forest management planning

Remsoft Inc.
332 Brunswick Street,
Fredericton, NB
Canada E3B 1H1

1-800-792-9468 or
1-506-450-1511

www.remsoft.com

Final Report

submitted to

New Brunswick Forest Research Advisory Committee

April 1994

Executive Summary

The purpose of this project was to compare the current forest management planning process in New Brunswick with an alternative based largely on computer software tools. Two New Brunswick Crown Licenses were used as case studies: forest classification schemes, yield estimates and assumptions about forest dynamics used in the study were identical to those used by each of the participating Licensees in their respective forest planning models. However, Remsoft staff used Woodstock to develop a strategic forest management schedule, Crystal to generate potential harvest blocks and Block to develop a spatially feasible block harvest schedule. All analyses were conducted on a Gateway 2000 4DX2-66V microcomputer running MS-DOS 6.2, with 16MB of memory and a 425MB hard drive using Doublespace compression.

To complete the analyses for both Licenses required six person-months of labor. This included time to obtain and process GIS map coverages for each License, to develop models and determine alternative solutions for them, and to write the final report. The bulk of the work lay in writing custom software to facilitate conversion of Licensee provided input files to, to update and obtain attribute and topological information from the GIS data files, to automate data manipulation between planning models and to present mapped solutions. With the system of procedures currently in place, we estimate that a Licensee familiar with the programs could complete the tasks undertaken in this study in approximately one week.

Results of the Woodstock runs showed that linear programming has significant advantages over simulation, in particular the ease with which outputs and activities can be constrained and the ability to readily control indirect outputs such as wildlife habitat. Overall, the Woodstock models yielded 9% to 22% increases in strategic allowable cut estimations over the baseline values provided by the Licensees. Moreover, mature conifer furbearer habitat requirements were met in all planning periods of the Woodstock analyses, unlike their baseline counterparts.

Despite a few shortcomings related to the ability to simultaneously address multiple harvest actions, Crystal appeared to work well, and compared to a manual approach, it was a vast improvement. Generating 10 alternative block layouts for each of 10 different blocking parameter specifications on one License required approximately 7 hours of processing using Crystal; on the other License, the time required was just over 22 hours. Depending on the License and the blocking parameters

used, Crystal was able to generate blocks for 50% to 95% of the area scheduled for harvest in the first 7 periods. Block performed adequately but required a significant amount of effort and custom programming to be able to use it efficiently. Despite its awkward input file structure and the inability to easily accommodate non-clearcut harvest prescriptions, Block produced good results in this study. For one particular block layout, the schedule developed by Block projected an average harvest just 2% lower than the harvest level of the Licensee's pre-block baseline analysis. Adjacency delays and maximum opening size constraints were never violated in any of the solutions, although License 4 was more prone to adjacency conflicts and thus a higher percentage of blocks remained unharvested by Block than was the case for License 4.

Whether or not Woodstock, Crystal or Block are used operationally in New Brunswick, it seems inevitable that software solutions will be adopted for harvest scheduling and blocking in the future. Several areas of potential problems in the future are identified which may become an issue as technology makes it faster and easier to explore more alternatives and include more constraints in forest planning models. Therefore, in addition to suggesting future modifications to Crystal and Block, recommendations include:

- *Conducting a benchmarking exercise for the two Licenses in this study using the approach taken by the FORMAN 2000 group.*
- *Providing Licensees with detailed attribute and topological data from the provincial geographic information database.*
- *Establishing consistent guidelines for planning procedures and articulating requirements and regulations in terms that are not dependent on a particular frame of reference.*
- *Implementing linear programming techniques as part of a broad-based planning methodology.*
- *Implementing joint planning activities for Crown and Licensee freehold lands.*

Background

Problem statement

Comprehensive forest management has always been difficult because of the magnitude of the problem. Early attempts concentrated on attaining relatively straightforward management goals such as forest regulation, from which the related goals of sustained yield and perpetual supply are realized by definition. Over time, the notion of the regulated forest has largely been discarded due to the dynamic nature of forest ecosystems and the inability to rationalize methods such as area control with the need to adapt changing social and economic demands. However, the goals of sustained yield remain, although they are much wider in scope than simply timber volumes. Thus the problem of forest management has become one of deciding what actions to perform on what part of the forest and when, to provide the desired benefits. Because some actions are incompatible with the production of some products, trade-offs exist for virtually all combinations of actions.

In New Brunswick, the Crown Lands and Forests Act requires licensees to produce an 80 year strategic plan, a 25 year management plan, and a 5 year operational plan. The purpose of the strategic plan is to define ways to meet long term management objectives, while the management and operational plans are location specific and details geographic locations of proposed activities. Currently, strategic plans are developed using a stratum or stand-type based approach for determining periodic harvest levels and management prescriptions.

However, since spatial factors (stand location, minimum and maximum harvest block sizes, maximum opening size, adjacency delay requirements) are not considered, following the stratum based harvest schedule is unlikely to produce a feasible management or operational plan. Instead, the stratum based harvest schedule is used as the basis for delineating sufficient numbers of harvest blocks to generate a block harvest schedule for a 25 year planning horizon. In the process of generating blocks, deviations from the stratum based schedule are necessary to comply with adjacency constraints and to even harvest flows. Once a feasible block harvest schedule has been found, it must then be validated by incorporating it into the strategic plan to ensure long term sustainability. If the resulting long term harvest level is unacceptable, adjustments to the block harvest schedule must be made until long term sustainability is ensured.

This three step process is implemented in New Brunswick using FORMAN+1 and FORMAN+2 for the strategic analysis and manual procedures for the remaining steps. Unfortunately, it is these steps which are the most data and labor intensive and several problems arise:

- It is not uncommon for licenses to spend over 60 person-days to achieve an initial harvest block schedule that meets all of the spatial, temporal and harvest flow constraints.

- Because of the high cost of finding and evaluating each solution manually, exploring alternatives is rarely undertaken and the impact of increased spatial and temporal constraints on wood supply is not addressed.
- The decision criteria used to obtain a particular schedule are usually not explicit enough to make the process repeatable.

Licenses used in the case study

Two Licensees agreed to be participants in this study: Valley Forest Products (VFP) (License 8) and Miramichi Pulp and Paper (MPP) (License 4). Each Licensee agreed to provide us with information used in the preparation of their most recent Crown Land Management Plan. This information included the class and yield information used in their FORMAN analyses, along with lists of stands comprising each class. The Department of Natural Resources and Energy (DNRE) provided us with 1988 vintage Forest Development Survey (FDS) base maps and watercourse buffer/deer wintering area overlay files in ArcInfo export format.

We decided to begin the study with License 8 since it was a more fragmented land base with a more heterogeneous forest classification and fairly complex management objectives. The rationale for this was to test the worst case scenario – if procedures could be developed for converting data for this License, then it would be relatively simple to do so for other Licenses with less complicated planning problems.

Valley Forest Products subdivided the License 8 forest area into different capability classes (unrestricted versus restricted access, softwood versus hardwood, even-aged versus uneven-aged), resulting in six individual FORMAN+1 models, plus individual models for deer wintering areas. Because of the need for regular flows of softwood and hardwood products, and to avoid the negative allowable cut effect due to subdividing the forest, we decided to build a single *Woodstock* model which would encompass all the different capability classes except the deer wintering areas.

In contrast, License 4 is largely dominated by softwood forest with large tracts of contiguous Crown land. Both the Licensee and sub-Licensees are primary softwood users and hardwood utilization is fairly low hence the management objectives tend to be rather consistent for all parties. In addition, License 4 has fairly large mature conifer furbearer habitat (MCFH) requirements compared to License 8, which is in a different wildlife zone with more emphasis on deer wintering areas. Unlike Valley Forest Products, Miramichi Pulp and Paper used a single FORMAN+1 model for the unrestricted land base plus additional models for deer wintering areas.

Valley Forest Products is the wood procurement agency for the Ste. Anne-Nackawic pulp mill, a hardwood mill which uses minimal amounts of softwood during processing. However, License 8 must also supply a number of sub-licensees, the majority of which are softwood users, primarily interested in spruce-pine-fir saw material and spruce-fir pulp. One of the major problems faced by VFP is maintaining a balance between the hardwood needed by the pulp mill, and the softwood fallout arising from harvesting in mixed wood stands. Simple maximization and/or constraining of a single product output leads to unacceptable fluctuations in the flow of other product outputs.

Miramichi Pulp and Paper manages two Crown Licenses in north-eastern New Brunswick. License 4 is comprised largely of lands bordering the upper Miramichi River basin. Unlike License 8, much of the forest is comprised of softwood species, primarily spruce and fir. In general, softwood pulp and log material is of primary importance with a much smaller demand for hardwood material.

The two License boundaries encompass roughly the same area: License 8 is distributed over 135 Forest Development Survey (FDS) map sheets, License 4 over 132. However, the Crown land portion of License 8 (126 157 ha) is substantially less than License 4 (356 871 ha); on License 8, much of the Crown land base is made up of Crown woodlots and small tracts, as opposed to License 4 which is essentially one large tract of contiguous Crown land. The average stand size on License 8, after overlaying watercourse and exclusion zone buffers, was somewhat smaller than the average on License 4 (2.8 ha and 3.2 ha respectively).

Hardware and software tools used in the case study

All analyses conducted in this study were performed on a Gateway 2000 personal computer with an Intel 486DX2-66 processor, 16MB of memory and a 425MB IDE hard disk. To perform the map import and overlay procedures, we used pcArcInfo Version 3.4D; other database manipulations were performed using FoxPro 2.0. In addition, numerous conversion programs and utilities were developed by Remsoft Inc. as a part of our own research and development program, including a polygon adjacency scan and utilities to draw and color code map sheets by harvest period.

Woodstock

Woodstock is an MS-DOS based forest modeling system developed by Remsoft Inc. to conduct forest planning analyses, including harvest scheduling. *Woodstock* models can be inventory projections, Monte-Carlo simulation models or linear programming (LP) models. Because of the very powerful constraint capabilities of LP, we decided to formulate the strategic wood supply analyses of both Licensees as linear programs. A brief overview of linear programming is given in the Appendix.

Crystal

Crystal (Walters, 1991) is an MS-DOS computer program, developed at the University of New Brunswick which is designed to allocate harvest prescriptions from a stratum-based harvest schedule to individual stands thereby providing a spatial configuration for part of a strategic management plan. *Crystal* allocates prescriptions on a stand by stand basis, and thus the blocks it generates are only precursors to final operational blocks. *Block*ing parameters such as block size and allowable deviations from the strategic schedule are controlled by the user. A brief overview of the *Crystal* algorithm is given in the Appendix.

Block

Block (Dallain, 1989), also an MS-DOS computer program developed at the University of New Brunswick, determines spatially feasible block harvest schedules under opening size, adjacency and harvest flow constraints. *Block* uses a Monte-Carlo integer programming (MCIP) algorithm to generate many alternative solutions to the block harvest scheduling problem. By retaining those feasible solutions with the highest objective function values, *Block* can generate very good, near optimal solutions in a relatively short time. Maximum opening size, adjacency delay and harvest flow constraints can all be specified by the user on a global basis as well as for individual management units and habitat zones. A brief overview of *Block* is given in the Appendix.

Methodology and results

Development of strategic harvest schedules using Woodstock

The automated blocking procedures used in the *Crystal* and *Block* programs require topological information about the arrangement of stands across the forest: what forest class each stand belongs to, what stands are adjacent to each stand, and the size of each stand. Since these data are readily available from GIS data files, we decided to combine the stratified forest information embodied in the Licensee's models with the stand level information provided in the forest cover and exclusion zone coverages from ArcInfo. Since the ultimate goal of the study was to automatically produce pseudo-blocks for block harvest scheduling, we decided to begin the strategic planning process with a spatially-referenced forest database to facilitate disaggregation later on.

Building the classification schemes

We examined the model input data provided to us by each of the Licensees to determine how they stratified their forests. Both used similar classification schemes (working group, site, silvicultural status and management unit), however Valley Forest Products divided the forest into several capability classes with a separate model devoted to each one. FORMAN+1 allows users to assign descriptive names to yield curves and forest classes, but these names need not be unique, nor do they have to correspond to one another. Instead, FORMAN+1 uses a numerical encoding format to match forest classes to yield curves. One disadvantage of this approach is that the codes themselves have little meaning, and the process of checking for errors in meaning is difficult. Therefore, we decided to build a classification scheme for the *Woodstock* models directly into the GIS database rather than simply convert the numerical encoding structure of the baseline models. There are two major advantages to this procedure:

- should a change in the classification scheme be necessary at the GIS level, none of the other steps to produce an area file for *Woodstock* would need to be modified; a new area file could be produced in minutes and the linkage to component stands would necessarily be maintained,
- in the future when a new round of management plans is implemented, the work done to associate ages and yield curves is saved; rather than go through the process of individually assigning

stands to forest classes, the information used in the previous planning cycle can simply be updated.

Using a combination of visual inspection and programming, we devised a consistent classification scheme for both Licensees, where unique 4 or 5 part labels were assigned to each forest class; each part of the label was designated a landscape theme. A custom program was written to modify the polygon attribute table (PAT) files in each coverage. New fields added to the PAT files included one for each landscape theme used to classify the forest, one to uniquely identify every polygon within the forest, and fields to assign block numbers and harvest periods later in the process. Once the block numbers and harvest periods are incorporated into the GIS database, it is trivial to produce maps of the block harvest schedules for visual inspection.

Accounting for watercourse buffers and exclusion zones

Next, we overlaid the forest coverages with coverages of watercourse buffer and wildlife exclusion zones. The overlay process combined the forest cover attributes with the buffer/wildlife attributes to create a new set of maps. Because many of the polygons in each coverage were not part of the productive land base, we decided to use a re-select operation to remove all of the ineligible stands to reduce the disk space requirements to store all the maps. Using a batch process to conduct the initial overlays followed by the re-select operation, it took more than 20 hours of processing to complete each License. The resulting coverages included all Crown land, with attributes from both the FDS and buffer coverages.

Model formulation - dynamics

After the new maps had been created, we merged all of the individual PAT files into a single attribute table. On the basis of landscape attributes and stand age, we used a database report writer to group the individual stands into unique classes and create a *Woodstock* analysis area file. Then, using a custom program written for the task, we converted the baseline input files to *Woodstock* format: yield curves were formatted in *Woodstock* format, harvest and silvicultural actions were defined using *Woodstock* syntax, and the baseline transition response file was converted to *Woodstock* syntax using the new classification system.

NOTE: *An overview of linear programming harvest scheduling models is given in the appendix.*

Once the major sections of the *Woodstock* model ere in place, we manually edited the files to remove redundancies and to structure the constraints and objective functions for the linear programming formulation. The conversion takes only minutes to complete, but the manual editing process can take a few hours, depending on the amount of streamlining desired. Once the procedures were finalized, converting a FORMAN+1 data set to a *Woodstock* model structure took roughly one day.

NOTE*: Both of the Licensees provided us with forest class files. In order to be certain that every stand was accounted for, with no possibility of duplication or omission, we embedded the landscape themes directly into the PAT files.*

The value in being able to conduct a forest management scheduling analysis within a single model framework should not be underestimated. Many of the difficulties associated with forest management planning arise because of competing resources and co-production of outputs. For example, it is difficult to produce hardwood pulp by clearcutting mixedwood stands without also generating softwood pulp. Conversely, the production of mature conifer furbearer habitat competes with softwood volume production since the same development types furnish both outputs. The only way around these difficulties is through trade-offs – judicious selection of activities and their timing to best meet multiple objectives. By separating the various components of the forest into discrete planning models, it is impossible to make these types of trade-offs.

Model formulation - LP constraints

The most difficult task in formulating the management problems of the two Licensees as linear programs was establishing constraints. The underlying principle of simulation models is trial and error: you tell the model what to do and it reports the results. The approach depends on the analyst's ability to deduce the impacts of various changes and implement controls which produce a desired result. With a linear programming approach, you tell the model what kind of solution you want and it reports the best means of accomplishing it. In effect, the roles of analyst and model are reversed, with the analyst providing the bounds for the solution space and the model determining the course of action.

Because of the long history of simulation modeling in New Brunswick, regulations and policy have come to reflect the modeling paradigm of FORMAN. For example, we were told that the minimum requirement for gross mature conifer furbearer habitat (MCFH) was based on the ratio of gross to net MCFH at the low point in the projected growing stock. We recognize that this determination is based on past experience with FORMAN projections and is a reasonable approach for this type of model. However, LP models require fixed quantities for constraints, either single numbers (i.e. $X \geq 30$) or fixed proportions of another quantity (i.e. $X \geq 30\%$ of Y) and a specific period for applying the constraint; the requirement for MCFH, as stated earlier, provides neither piece of information. Furthermore, it is possible to formulate a LP model where the growing stock is at a minimum in any desired period, or one which does not exhibit a dip in the growing stock at all.

NOTE: *A nondeclining yield constraint sets up a series of linkages between planning periods where the output level of any period must be greater than or equal to the output level of the previous period.*

In order to produce harvest schedules which were reasonable approximations of those produced by the Licensees, we constructed *Woodstock* models with constraints which we felt captured the intent of provincial regulations and Licensee objectives. To ensure that silvicultural activities were maintained at required levels over the planning horizon, we imposed a perpetual timber harvest constraint in the final planning period. Without such a constraint, the optimal solution will produce just the amount of inventory in the last periods to sustain the required harvest level. However, such inventory levels would not likely result in sustainable harvests beyond the end of the planning horizon.

By examining the solutions found by the Licensees, we were able to determine a minimum ratio of gross-to-net MCFH area for a specific period. To approximate the wildlife habitat requirements on each License, we established two constraints. The first constraint guaranteed that the area of MCFH-eligible age classes within the specified zones did not fall below initial values for the first seven planning periods. In all subsequent planning periods the gross MCFH area was constrained to be at least a fixed area: this minimum area was determined by examining the gross MCFH area in the period where the growing stock was at a minimum in the baseline analyses.

Valley Forest Products expressed a need to control the flow of both primary and secondary products. Since FORMAN+1 does not provide a means of directly controlling secondary product flows, the License 8 forest was subdivided into capability classes based on the predominant product harvested from each forest type. This approach allows you to set the predominant output as the primary product and control it, however all other outputs remain as fall-out products. The net result may be less variation overall in periodic output levels, but there will still be some variation due to fallout products. Furthermore, a negative allowable cut effect can be expected because of the subdivision of the land base.

For the License 8 model, we implemented nondeclining yield constraints on total harvest volume, softwood pulpwood/logs, and mixed-hardwood pulpwood. Other product flows were not directly constrained but because they were components of total harvest volume the harvest levels of these products were bounded by the non-declining yield constraints. For the License 4 model, nondeclining yield constraints were placed on total softwood volume and total hardwood volume. For both Licenses, we formulated objective functions which represented the major product demands from the License. For License 8, the Licensee requires hardwood material but the sub-Licensees are primarily softwood users so the objective function maximized first period harvest of total softwood and hardwood volume from even-aged and uneven-aged silvicultural prescriptions. For License 4, both the Licensee and sub-Licensees are primarily softwood users so the objective function maximized first period harvest of total softwood volume only. Valley Forest Products also projected significant harvest volumes from uneven-aged management. Since these projections originated in FORMAN+2, we simply took the results of the FORMAN+2 runs and coded the outputs as time dependent yields in *Woodstock*. Although the *Woodstock* model had the option of implementing the unevenaged management prescriptions, it could not change the harvest levels arising from these prescriptions. Therefore, the unevenaged volume components are exactly the same as those reported by Valley Forest Products.

NOTE: *The perpetual timber harvest constraint assumes that if the ending inventory is at least equal to the average inventory over the entire planning horizon, then a regime of harvest and silviculture similar to the one used during the planning horizon should be*

While it would have been possible to maximize total volume over the planning horizon, this would have placed as much emphasis on harvest volumes from the last planning period as the first. Furthermore, the first period harvest may have been reduced so that additional volumes could be harvested in later periods. Neither of these outcomes reflects Crown or company objectives for forest management planning and so we limited the objective function to the first period.

In keeping with Provincial policy on silviculture, we did not place any constraints on silvicultural activities. DNRE regulations stipulate that the Licensee must perform the level of silviculture which will maximize the allowable cut effect. With an objective function to maximize first period harvest and concurrent flow constraints on the outputs being maximized, the *Woodstock* models determine the maximum allowable cut effect by default. Furthermore, only the silviculture which contributes to an increase in first period harvest is performed; additional silviculture that could increase inventory but not increase the first period harvest is not done. Although LP models are efficient in finding this type of solution, the marginal cost of producing this wood may be very high.

Solving the Woodstock models

Although the land base of License 4 is significantly larger than License 8, the complexity of the License 8 model resulted in a LP matrix more than twice the size of the License 4 matrix. Furthermore, whereas the License 4 LP solved in about 30 minutes on our computer, the License 8 LP required nearly 5 hours to solve on the same machine. Much of the difference in solution time between the two is due to the greater number of constraints present in the License 8 model; LP solution time is particularly sensitive to the number of constraints.

Results of the Woodstock models

Determining the model structure, developing the conversion and utility programs, updating the GIS coverages and producing the final *Woodstock* model required about four weeks time. However, now that the procedures have been developed, it should be possible for users familiar with both modeling approaches to convert a FORMAN+1 analysis to *Woodstock* within a day or two.

In the case of License 8, the *Woodstock* model projected an allowable cut significantly higher than the allowable cut reported by Valley Forest Products using FORMAN+1. Linear programming models are particularly adept at capitalizing on trade-offs among different stand types and across planning periods, a feature of particular value in the highly constrained *Woodstock* model for License 8.

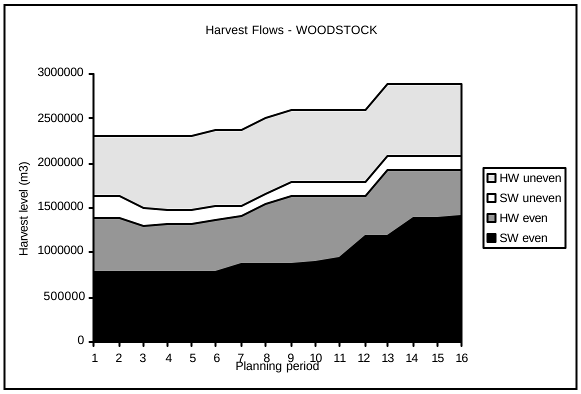


Figure 1. Projected harvest levels from Woodstock strategic model for License 8.

The evenaged hardwood component includes birch and poplar products which were not subject to flow constraints. These products are the cause of the minor variation in the harvest flows of evenaged hardwood. However, the harvest profile reflects a general increasing harvest level over time, particularly for evenaged softwood products. The unevenaged component harvest flows of the *Woodstock* model are identical to those of the License 8 baseline models.

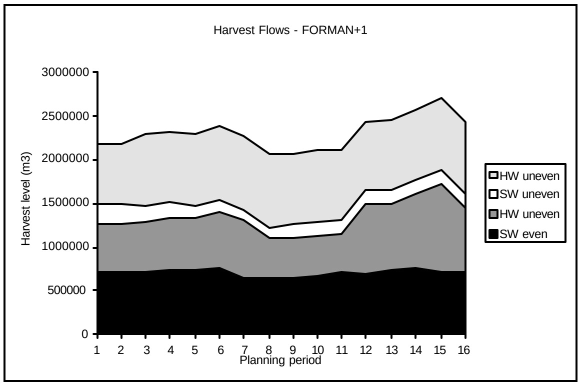


Figure 2. Projected harvest levels from Valley Forest Products' baseline models for License 8.

Although even-aged softwood products exhibit relatively little variation period to period, evenaged hardwood products vary a great deal. Furthermore, despite a trend toward increasing harvest levels in later periods, there is a significant lapse in this trend in the middle periods. In addition, the evenaged softwood component does not exhibit the increases in allowable cut of the *Woodstock* model

The *Woodstock* model reported an annual harvest in the first period of 279,000 m^3 from the evenaged capability classes, whereas the baseline models projected annual harvests in the first period of 255,000 m^3. A comparison of the inventory profiles of the *Woodstock* and baseline models showed a general decline in inventory levels over time in the baseline runs, while the *Woodstock* model maintained more than double the level of inventory of the baseline models, despite harvesting more wood.

Although it is not visible in the graph, small amounts of hardwood log volume are produced in later periods. Note also the shift toward softwood pulp production after period 5. ➥

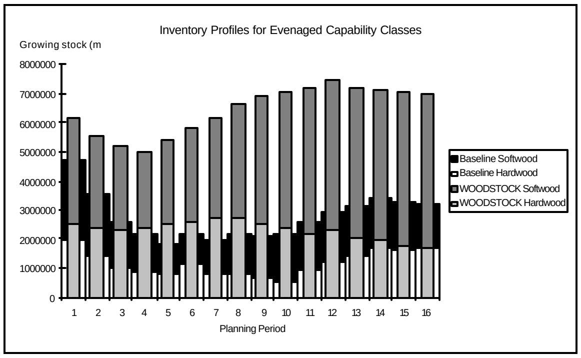


Figure 3. License 8 inventory profiles projected by Woodstock and baseline models. What is particularly striking about this figure is that the Woodstock model was able to retain more than twice the inventory of the License 8 baseline models, while harvesting more wood.

The harvest profile for License 4 was very different from License 8. Although the same harvest flow constraints were used, the flow of total softwood and total hardwood components were strictly even, although there was a shift toward increasing softwood pulp and decreasing softwood logs in later planning periods.

***NOTE**: An overview of the algorithm used in Crystal is given in the appendix.* ➥

The allowable cut projected by the *Woodstock* model was approximately 787,000 m^3 annually; as compared to an AAC of 647,000 m^3 using the baseline strategy reported by Miramichi Pulp and Paper. The MCFH requirement was satisfied in all planning periods for the *Woodstock* model, whereas it was not met in periods 14 through 16 in the License 4 baseline projections.

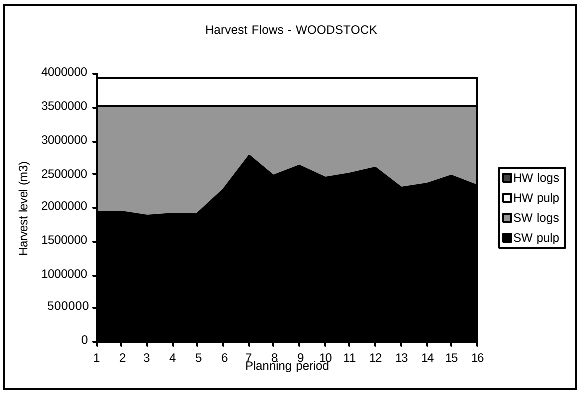


Figure 4. Projected harvest levels from Woodstock strategic model for License 4.

For both Licenses, the *Woodstock* models yielded higher allowable cuts than the corresponding baseline analyses performed by the Licensees. Furthermore, the optimal solutions found using *Woodstock* met all planning requirements that were formulated as constraints; the baseline models of both Licensees appeared to project shortfalls in one or more outputs during the planning horizon.

Developing harvest blocks using Crystal

Harvest treatment tables

The harvest schedules developed with *Woodstock* for each License were the basis for blocking with *Crystal*. The report writing capabilities of *Woodstock* were used to write an analysis area report for the first 7 periods of the planning horizon. These ASCII files were imported into xBASE format data files, one for each License. Only harvest prescriptions (commercial thinning or clearcutting) were maintained in the data file and all other actions were deleted (planting, spacing, senescence). Silvicultural prescriptions were not blocked because of the inability to predict sites requiring treatment. Since eligibility for treatment in the strategic model was based on a forest-wide sample rather than stand-level attributes, any stand within an eligible development type may or may not actually require treatment. Therefore, we assumed that Licensees would implement treatment where needed.

The *Crystal* algorithm was designed only for single entry harvest prescriptions. Although commercial thinning is not a single entry harvest, none of the treated development types were scheduled for second entries during the seven period planning horizon and thus the commercial thins could be accommodated. Two-pass harvests, however, could not have been easily accommodated in *Crystal* or *Block*. In the initial runs of the License 8 *Woodstock* model, a limited amount of two-pass harvesting was also selected. However, two-pass harvests did not contribute a large amount of volume, and because Valley Forest Products did not implement two-pass harvests in their baseline runs, and because of the complex workarounds that would have been required to use *Crystal* and block, we modified the *Woodstock* model to exclude two-pass harvests for License 8. In the License 4 model, two-pass harvesting was never selected.

Adjacency tables

The pcArcInfo topology structure can provide information on stand adjacencies within a map coverage, but cannot provide adjacency information across map boundaries. Also, the map sheets provided by DNRE had not undergone edge-matching, a process which guarantees common boundaries between adjacent map coverages. Since pcArcInfo provides no librarian functions available in the workstation versions of ArcInfo, we developed a custom program to determine stand adjacencies within and across map boundaries. The output of this program was imported into a xBASE file, duplicate records were removed and then the file was restructured as a double entry list. The process of generating the adjacency table had to be done only once for each License, and required less than an hour to complete on our computer.

Eligibility tables

The eligibility table was simply the common attribute table generated earlier when the map overlays were processed. The only modification required for *Crystal* was to sort the file on the basis of map and stand number. Preparation of a *Crystal* input data set required no more than a couple of hours, including time to generate the adjacency table.

Setting blocking parameters

One of the objectives of the study was to determine how well *Crystal* worked under different planning conditions. The two Licenses in this study had very different forest structures and management objectives. To retain a degree of comparability, we decided to apply the same sets of blocking parameters to both Licenses. Although we did not know what minimum block size would be acceptable to each company, we tested minimum block sizes ranging from 5 up to 25 hectares in size with target block sizes double the minimum. To determine the impacts of timing choice deviations within blocks, we established allowable deviations in timing choices were ±2 periods for type 1 stands , ±1 periods for type 2 stands and ±3 periods for use in the cleanup routine for the first set of runs. The second set of 5 runs used the same range of minimum block sizes, but allowed ±4 period deviations for type 1 stands, ±2 period deviations for type 2 stands, and ±5 period deviations in the cleanup routine. In all, 100 block configurations were generated for each License.

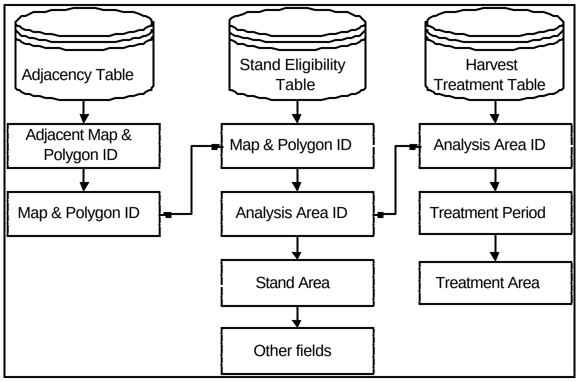


Figure 5. Relational structure of Crystal input files.

Allocating blocks

To accommodate both commercial thinning and clearcut prescriptions in *Crystal*, we allocated commercial thinning to blocks first. Because the area to be allocated to commercial thins was far less than clearcuts, we did not see this as a problem; had the area of commercial thins been comparable to the clearcut area, the decision as to which prescriptions to allocate first would have required more thought. We used a 10 ha minimum block size for all commercial thins and did not allow any timing deviations whatsoever for either License. Still, *Crystal* was able to allocate virtually all of the area scheduled for commercial thin prescriptions. Because there were no timing choice deviations and only one minimum block size used for commercial thins, we only ran *Crystal* once for each License, retaining the highest scoring block layout.

The time required to generate 10 alternative clearcut block layouts for License 8 (35 - 52 minutes) was far less than that required for License 4 (123 - 145 minutes), but the variation between runs was much higher for License 8 than License 4. In total, to generate 100 different block layouts for License 8 on our computer required 7 hours, 16 minutes; the 100 different block layouts for License 4 required 22 hours, 2 minutes.

Results of the Crystal block allocation

Crystal was much more successful at allocating larger blocks (20 or 25 ha minimum) on License 4 than on License 8; for the small blocks, there was little difference. In both cases, allowing more timing choice deviations enabled *Crystal* to allocate more of the area schedule for harvest. Furthermore, as the minimum block size increased, the proportion of scheduled area *Crystal* was able to successfully allocate fell, but at a faster rate on License 8 than License 4. Overall, on License 8 delineating harvest blocks much larger than 10 ha is problematic because significant amounts of area scheduled for harvest remain unallocated.

Each block generated by Crystal is shaded using a random color; unallocated areas are white. However, adjacent blocks may represent identical harvest prescriptions and timing choices (see Figure 9).

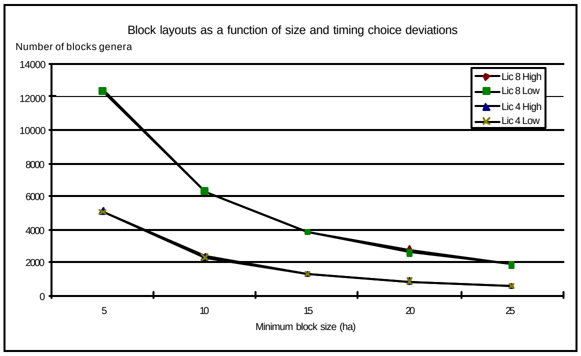


Figure 6. Area successfully allocated by Crystal for each License under various blocking parameters.

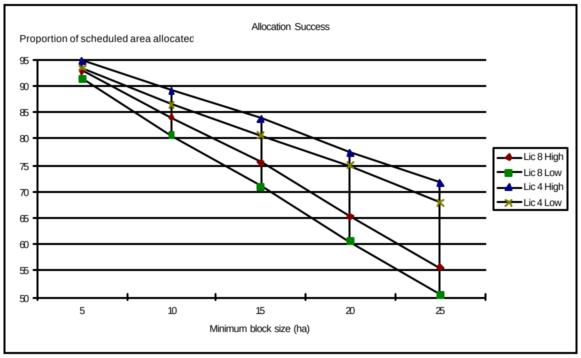


Figure 7. Number of blocks generated by Crystal for each License under various blocking parameters.

There was a great deal of variation in solutions across block runs (different minimum block sizes or deviations permitted), but little variation within runs. Typically, the overall score values for individual solutions (a measurement used to penalize large timing choice deviations) and the number of blocks allocated were very similar. For example, the number of blocks allocated on License 8 with a minimum block size of 5 ha using low deviations ranged from 5050 to 5073 with an average of 5061 blocks.

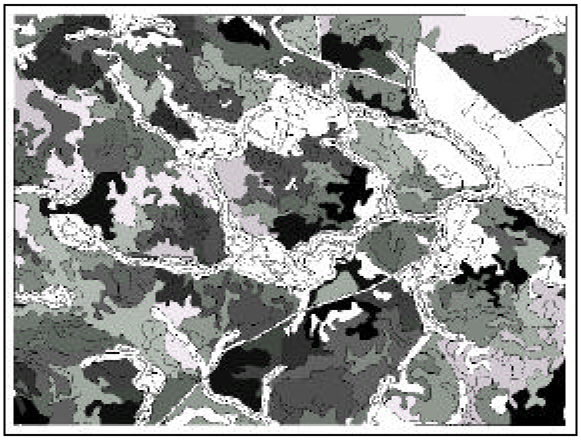


Figure 8. License 4 map sheet showing individual Crystal blocks.

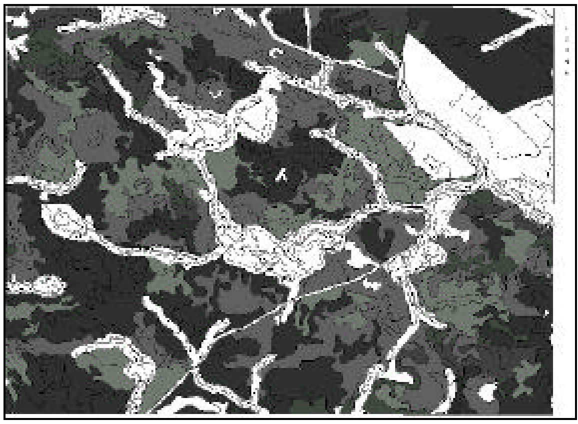


Figure 9. Preferred harvest times for individual blocks on License 4. The various shadings on this figure represent the final harvest periods for blocks. Where two or more blocks may be assigned the same harvest period, they will appear as a uniformly shaded opening. Stands not eligible for harvest are white.

Developing block harvest schedules using Block

Preparing the Block input files

Four different block layouts for each License were selected for scheduling. Each time *Crystal* was run, the best solution found thus far was saved, as well as information on the number of blocks generated, the overall score values, the proportion of area allocated using a specific timing choice deviation, area impossible to allocate and area left unallocated. The solution files are stored as dBASE IV files and detail the component stands for each block, size of each block, and block adjacencies.

Although *Crystal* provides most of the information required by *Block*, it is not in an appropriate format to be used directly. Furthermore, *Block* requires block volume estimates rather than stand type estimates of volume. To assist in producing a properly formatted *Block* input file, we wrote two custom programs. The first program reads the *Woodstock* input files to obtain yield and analysis area information. It then produces an intermediate file, which details per hectare estimates of previously defined outputs for each analysis area defined in model. The second program uses this intermediate file, along with the solution files produced by *Crystal* to calculate block volume estimates and write out a properly formatted *Block* input file. Finally, the block information for the commercial thin blocks is manually added to the input file. With the assistance of the conversion programs, developing a *Block* input file takes just minutes.

Like *Crystal*, *Block* was designed only for single entry harvest prescriptions. However, the maximum opening size and adjacency delay parameters can be different for different management units or habitat zones. Because the commercial thins are not considered openings and the final harvest of these areas does not occur during the planning horizon, we separated the two types of blocks using the management zone option. This allowed us to apply a maximum opening size of 100 ha and a 10 year harvest delay for clearcut blocks without restricting commercial thin blocks whatsoever. Also, volume obtained from both harvest prescriptions contribute to the volume objective, which would not be possible with separate runs for each.

NOTE: *An overview of the algorithm used in Block is given in the appendix.*

Block runs

For each run, we restricted the availability of commercial thin blocks to the periods in which they were originally scheduled by *Woodstock*. Clearcut blocks could be scheduled during any of the 7 planning periods. To obtain relatively good solutions, an iterative approach was followed. For the first run, we applied no limits on individual product flows and generated 100 feasible solutions. Then, we examined the best solution found, and noted what the lowest harvest level was for each product over the planning horizon. We then ran *Block* again with lower limits on each product set to the minimum values found in the previous run. By applying the same procedure 3 or 4 times, we quickly found appropriate lower limits which would yield approximately one feasible solution for every 100 attempts. Then, we ran *Block* once more, using the final lower limits for each product, to generate 100 feasible solutions. The best 3 solutions from each run were retained. In most cases, generating a final solution set for a particular block layout required about an hour.

The final step in the process was to match up the final harvest periods for each block from the *Block*-generated harvest schedule with the individual stands in the master polygon attribute table. A custom program was written to perform this function, which simply updated the period field with the harvest period selected by *Block*. *Block*s left unharvested by *Block* were assigned a harvest period of zero.

Results of the Block runs

For License 8, the 10 ha minimum block layouts yielded about 330 blocks as opposed to about 115 for the 20 ha minimums. Unlike License 4, only one or two blocks at most were left unharvested, regardless of minimum block size. Again, the high deviation layouts yielded higher average harvests than the low deviation layouts.

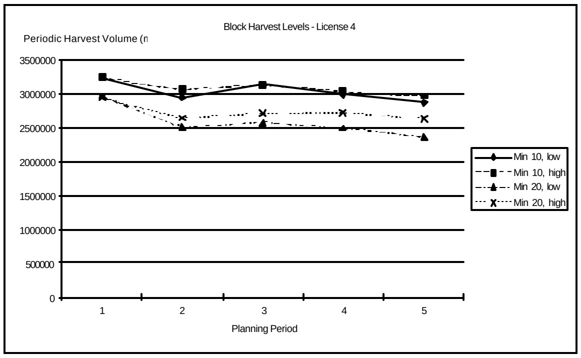


Figure 10. Block developed spatially feasible harvest schedules for License 4.

A comparison of the results from the various runs showed that smaller minimum block sizes in *Crystal* allow more of the schedule area to be allocated to blocks than larger minimum block sizes, with concomitant increases in average harvest levels in the corresponding *Block* runs.

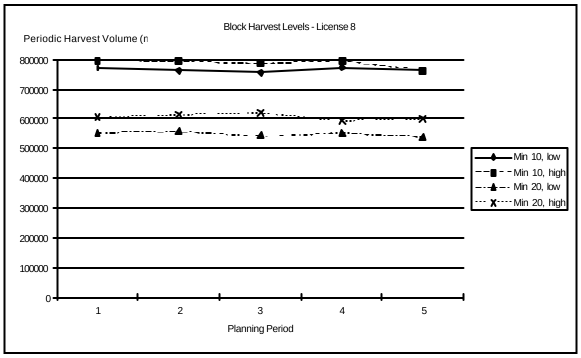


Figure 11. Block developed spatially feasible harvest schedules for License 8.

Mapped solutions quickly illustrate the differences in allocation success between the two Licenses. In particular, note the fragmentation in the land base, and the number of watercourse or wildlife buffers present on the map sheets from the two Licenses.

For License 8, the 10 ha minimum block layouts yielded about 330 blocks as opposed to about 115 for the 20 ha minimums. Unlike License 4, only one or two blocks at most were left unharvested, regardless of minimum block size. Again, the high deviation layouts yielded higher average harvests than the

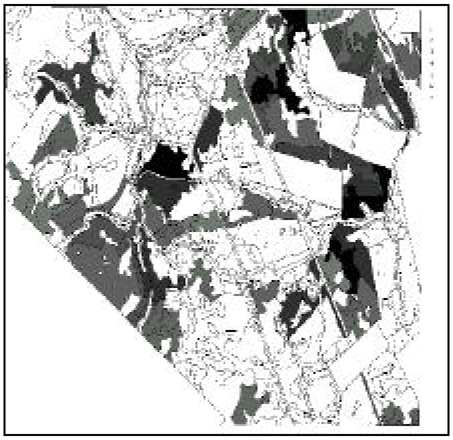


Figure 12. Scheduled block layout for a single map sheet from License 8.

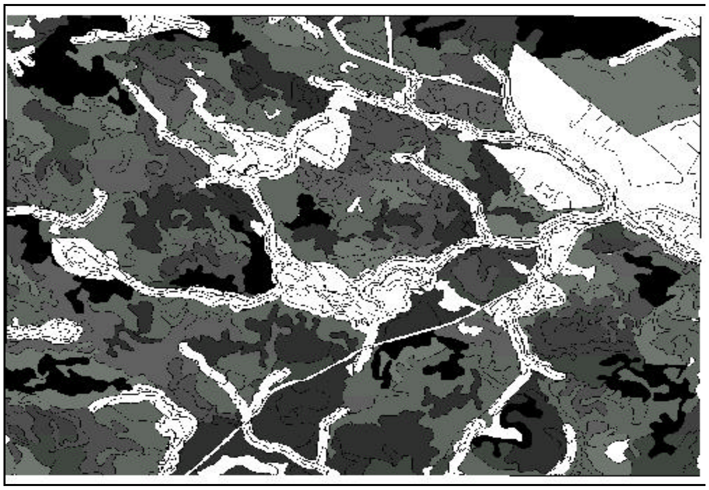


Figure 13. Scheduled block layout for a single map sheet from License 4.

Presuming that the blocks could be harvested as scheduled, the block harvest schedules for each License resulted in substantial decreases in AAC as compared to the optimal forecasts from *Woodstock*. For License 4, the decreases ranged from 19% to 30% whereas the decreases for License 8 were between 36% and 56% depending on the minimum block size used and the degree of timing choice deviations allowed.

Using a 10 ha minimum block layout generated by *Crystal* on License 4, the spatially feasible AAC produced by *Block* was 635 500 m3 per year; compared to the pre-blocked AAC from the preferred strategy developed by Miramichi Pulp and Paper staff which was 647 000 m3 annually (a difference of 2%).

The variation in block size period to period was relatively constant for every *Block* schedule: for example, using a 10 ha minimum layout from License 4, the average block size for the seven periods ranged from 23.4 ha to 25.3 ha with no violations of the adjacency constraint (see Figure 14).

License 4 appears to be more prone to adjacency conflicts than License 8. In all cases, the number of blocks left unharvested by *Block* was proportionally higher on License 4 than License 8. We presume that this is so because License 4 is far less fragmented than License 8, necessarily increasing the likelihood of adjacency conflicts.

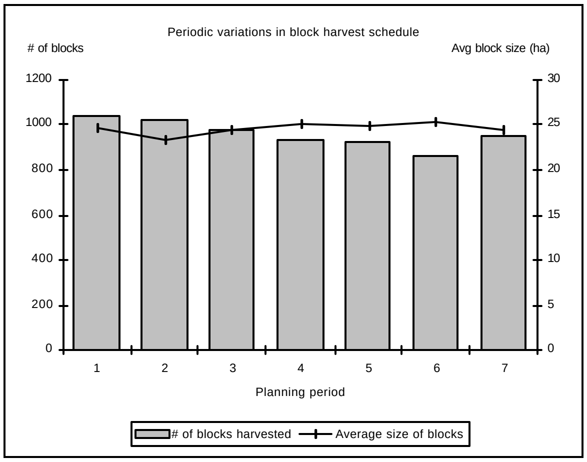


Figure 14. Variation in average block size and number of blocks harvested for a 10 ha minimum layout on License 4.

Based on the results found using *Woodstock*, *Crystal* and *Block*, and the ongoing research into related approaches, it seems inevitable that software solutions will be adopted for harvest scheduling and blocking. Even as technology makes it possible to explore more alternatives and consider more variables in forest planning models, those same capabilities can give rise to several areas of potential problems. These issues are discussed in this section.

Issues

Strategic harvest scheduling issues

Planning horizons

In many jurisdictions, the convention for setting the planning horizon is to at least double the average rotation length. The rationale for this is to ensure that by the end of the planning horizon only wood from regenerated stands is contributing to the allowable cut. Since long term sustained yield (LTSY) by definition is based solely on expected regeneration volumes, the final period harvest is usually a good indication of LTSY.

The data presented in the figure comes from a Forest Management Area in northern Alberta where average rotations range from 80 to 110 years. The objective function maximized first period harvest subject under non-declining yield constraints. With a sufficiently long planning horizon, the harvest level and the LTSY would be equal but the differences shown here are due primarily to surplus inventory – the models with shorter planning horizons liquidate the surplus at a faster rate thereby increasing the cut.

In New Brunswick, the required planning horizon is 80 years, which is less than two average rotations on License 4 and License 8. Short planning horizons generally exhibit higher AAC and lower LTSY values than longer planning horizons (see Figure 15). Because the existing inventory can be liquidated in a shorter time, allowable cuts are usually higher for short planning horizons; as the planning horizon lengthens, the existing inventory must last longer, until finally regeneration volumes are sufficient to sustain the harvest.

For the hypothetical forest depicted in Figure 15, an arbitrary ending inventory of 7 million m3 (approximately 50% of initial inventory) was required in the last planning period. As the planning horizon increases in length, more regeneration volume contributes both to the allowable cut and to the inventory. Beyond 24 periods increasing the planning horizon makes little difference – in other words, the allowable cut is essentially the same as the long term sustained yield.

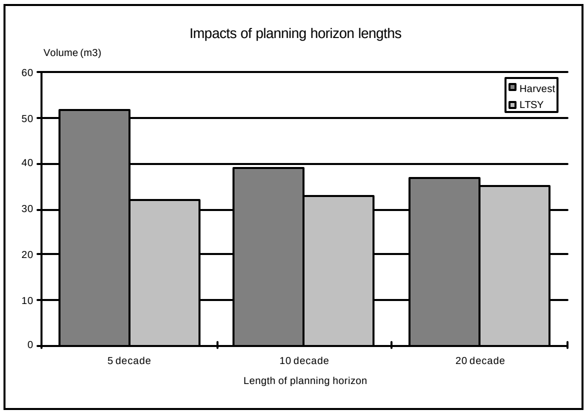


Figure 15. Changes in AAC and LTSY due to planning horizon length.

The effects of shorter planning horizons can be offset somewhat by imposing an ending inventory requirement. The perpetual timber harvest constraint works to counter inventory liquidation and thus ensure harvests beyond the end of the planning horizon. Although it is not a perfect substitute for longer planning horizons, it does tend to lower the estimated AAC closer the LTSY for the forest. Otherwise, there is a very real possibility of overestimating sustainable harvest levels (refer to Figure 16).

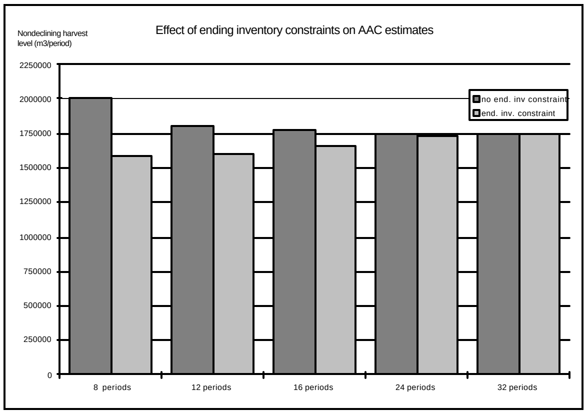


Figure 16. Allowable cut estimates for various planning horizon lengths with and without ending inventory constraints.

Although it is true that a new wood supply analysis every 5 years will correct for overestimates in harvest level, there will likely be more variation in allowable cuts by doing so. One advantage of using longer planning horizons and ending inventory constraints to estimate AAC's linked to long term sustained yield is for evaluating Licensee management performance. For example, a License which demonstrated maintenance or an increase in long term sustained yield would generally be considered in compliance with provincial management objectives; on the other hand, falling LTSY estimates would indicate potential problems.

Allowable cut effect

Current New Brunswick policy requires all Licensees to perform basic silviculture at levels which will maximize the allowable cut effect (ACE) – the immediate increase in harvest due to changed assumptions about future productivity or utilization standards. In attempting to comply, Licensees using FORMAN+1 have tried various silvicultural regimes to find the combination which yields the highest AAC. However, simulation models are rather poor at finding marginal increases in output and except in relatively simple models, linear programming models are better able to capitalize on silvicultural treatments and report substantially higher allowable cut effects than corresponding simulation models (Jamnick, 1990).

Using a linear programming formulation for License 8, we were able to find determine a silvicultural regime which maximized allowable cut effect. However, the cost of this regime is substantially higher than the one proposed by Valley Forest Products and yielded only a 9% higher harvest in the first planning period. Despite the fact that LP models nearly always yield higher allowable cut effects, the question remains whether such gains are economically viable.

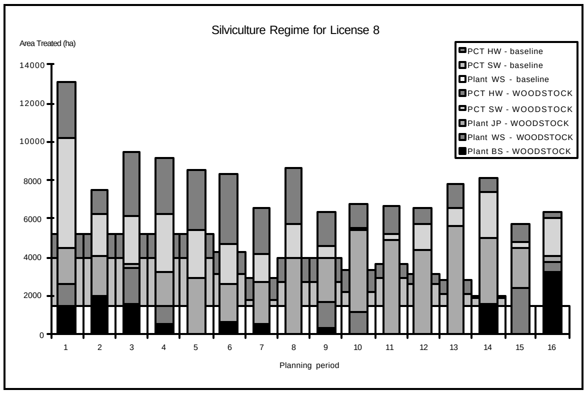


Figure 17. Silviculture regimes for License 8 using Woodstock and baseline planning models.

Not only does the optimal silvicultural regime result in far larger treatment areas and associated costs, but the fluctuations in treatment period to period are likely unacceptable from an operational standpoint. Although constraints on treatment could smooth out these fluctuations, they do not address the root problem of the ACE policy itself – that it is not economically justifiable, at least for basic silviculture. A more justifiable policy might be to set basic silviculture budgets at the point of diminishing returns, where further investments no longer increase at the rate of investment. Additional silvicultural investment to improve product quality or to increase future productivity would be the decision of the Licensee.

Wildlife habitat

The mature conifer furbearer habitat (MCFH) objectives require contiguous areas of mature softwood types. The current policy is to identify such areas and preserve them for as long as possible. Thereafter, new areas will need to be identified to replace those that are no longer suitable. The problem with the current modeling approach used on Crown land is that no attempt is made to address the contiguity issue in future periods. Setting aside areas for the present excludes them from harvesting, but no attempts are made to locate harvests in specific areas to create contiguous areas of forest with similar age and species composition. Without some form of zone-based spatial constraints it is doubtful that suitable habitat areas will be available at the appropriate times in the future. Moreover, the current policies on maximum opening size and adjacency constraints promotes even further fragmentation of the forest.

An automated blocking algorithm like *Crystal* depends on a strategic harvest schedule to determine eligible stands for harvest in each period. *Crystal* is only able to work within parameters established by strategic harvest schedule and if that schedule reflects dispersed harvesting and fragmentation, so will the blocking strategy generated by *Crystal*. The only way to counter this and concentrate harvesting would be to deviate from the strategic harvest schedule, the exact opposite of what *Crystal* was designed to do.

There are aspects of the current planning procedures related to habitat management which have strong implications for harvest scheduling. First, the eligibility windows for MCFH typically encompass the point of the yield curve where mean annual increment (MAI) is culminated. A stand which could be applied to the MCFH requirements can only be harvested after it is in decline to maximize its membership in the eligibility window; in many cases, the eligibility window extends beyond the usual operability window resulting in the complete loss of that stand for harvesting purposes. The result is that the objective of volume maximization is directly at odds with fulfillment of the habitat objective. Since both cannot be simultaneously attained, some form of trade-off is needed and the analyst needs to determine its magnitude.

Although the *Woodstock* model consistently projected less MCFH area than the baseline model up to period 11, it always met the minimum requirement, which the baseline projections failed to do in the last four planning periods.

Because the *Woodstock* models were able to make trade-offs across planning periods and among silvicultural treatments, the reductions in AAC due to MCFH requirements could be minimized. In general, the LP solver selects an appropriate harvest and silvicultural regime to just meet the MCFH requirements and nothing more. In contrast, FORMAN+1 models are rule-based and cannot make trade-offs across planning periods. Overall, the solutions it produces are inefficient: excess MCFH is produced in some periods, shortfalls occur in others.

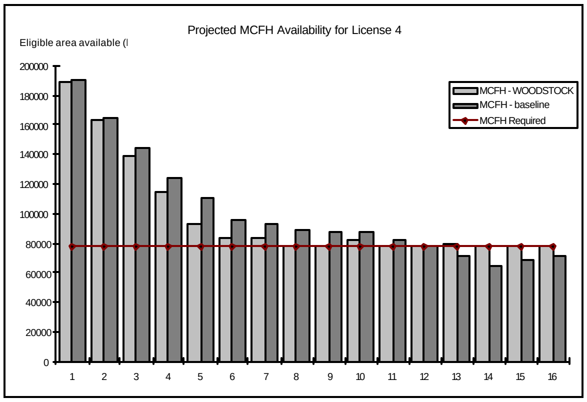


Figure 18. Mature conifer furbearer habitat availability regimes for License 4 using Woodstock and baseline planning models.

Product flows

All simulation models are limited in their ability to constrain outputs; for a given License, it is possible to constrain individual products for desired harvest flows (particularly valuable for Valley Forest Products) but one can also address issues which cannot be easily attempted using simulation models.

Although the Woodstock model consistently projected less MCFH area than the baseline model up to period 11, it always met the minimum requirement, which the baseline projections failed to do in the last four planning periods.

For example, most of the sub-Licensees are sawmill operators dependent on log material from each License but in most cases it is assumed that log volumes will decline over time with no guaranteed minimum supply. Because FORMAN+1 allows direct control of primary volumes only, it is difficult to establish minimum output levels for other products, except by indirect means such as changing harvest rules. However, with an LP based harvest scheduling model, there is no reason why minimum quantities of products like sawlogs cannot be maintained. The total allowable cut may decrease because of the longer rotations required to produce sawlogs, but sub-Licensees could be guaranteed a reliable supply of material.

Harvest blocking issues

Over the course of this study a key issue was the relative importance of strategic and tactical goals. A strategic forest plan indicates what stand types to harvest in a particular order to achieve a high allowable cut, hence the goal is largely to maximize volume. A tactical forest plan indicates the location of specific harvesting and silvicultural activities and the goal is generally to minimize costs. In order to link the two goals into an overall management plan, compromises need to be made, however, which goal takes precedence? Do you sacrifice allowable cut so that you can have large harvest blocks which reduce harvest costs, or do you accept smaller blocks with higher harvest costs to meet the long-term objective of a sustainable harvest into the future? The tradeoffs are real, but because of the difficulties associated with manual blocking, alternatives are rarely attempted and so trade-off analysis cannot be done.

Nature of the land base

The two Licenses selected for this study are very different, not only in terms of management goals, but in the very land bases themselves. Both Licenses cover approximately the same number of map sheets (132 for License 4 and 135 for License 8) but the configuration of those map sheets and the distribution of Crown land within the License boundaries are very different (Figures 19 and 20).

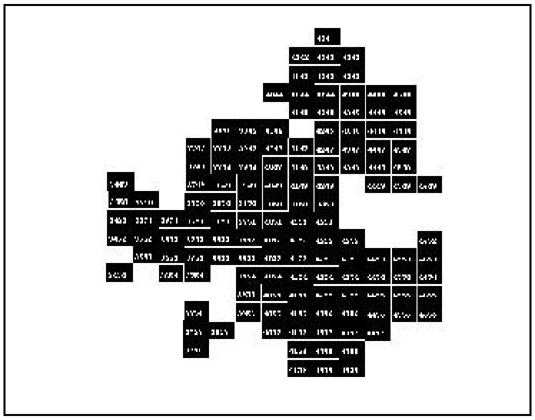
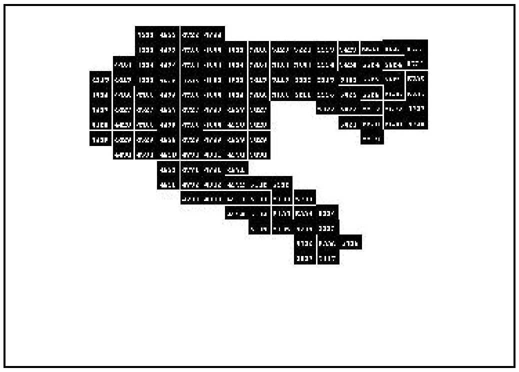


Figure 19. Map sheet boundaries for License 8 (top) and License 4 (bottom).



License 8 encompasses large areas of private land, mostly small woodlots interspersed among the Crown land holdings resulting in about one-third the Crown land of License 4 which is essentially one large contiguous land base.

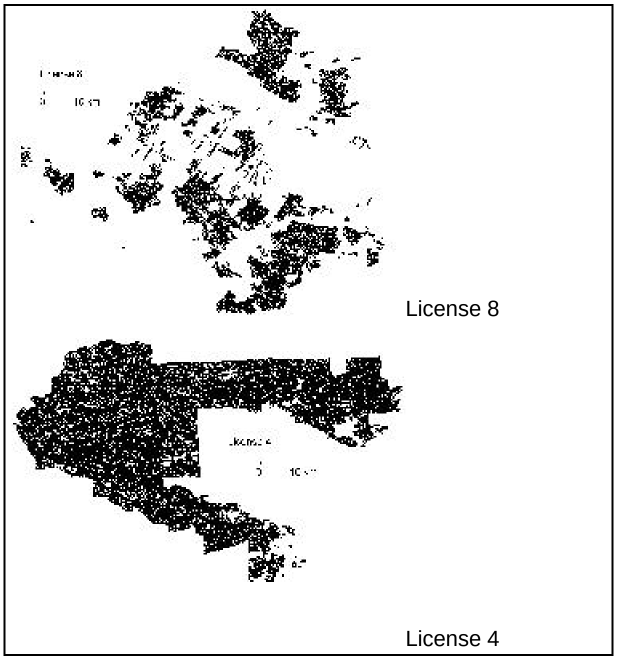


Figure 20. Crown land bases for License 8 and License 4.

From the stand point of simple geometry, it is obvious that for a given polygon and size, there are far more possible arrangements on License 4 than License 8. For the many Crown woodlots on License 8, there is probably only one way to configure harvest blocks on them that does not result in even smaller fragments to be harvested in future periods. While *Crystal* would be expected to have far more difficulty finding feasible harvest blocks on License 8 than License 4 due to the relative lack of contiguous land available for blocking, the isolation of individual harvest blocks means that adjacency delays would be expected to be less problematic on License 8 than License 4. The results presented earlier bear out these assumptions.

One means of circumventing the problem of fragmentation on License 8 would be to undertake joint planning for both the Crown and company freehold land. It would still be possible to recognize different management requirements and objectives for the two land bases in a single planning framework, but significant advantage would be gained. First, some form of allowable cut effect would be due simply to complementary forest structures, and second, reducing the fragmentation of the land base would present to *Crystal* a greater range of possibilities for generating harvest blocks.

Blocking parameters

Anyone using *Crystal* must specify a minimum block size parameter which the algorithm uses to decide whether a potential block should be retained or discarded. However, in our discussions with various personnel associated with this study, we found that the concept of minimum size was not universally adhered to. In some cases, a minimum block size was truly a lower bound, and blocks smaller than this size were never acceptable. In other cases, the specified minimum was more of a desired minimum and although blocks smaller than this were not desired, they might be acceptable in a few instances.

In many ways, deciding upon a minimum block size is an economic decision. In general, a harvest block must contain a certain amount of volume to justify the fixed costs of moving harvesting equipment to the site. However, if a small block is located close by to another larger block, it may be feasible to simply drive a skidder or feller-buncher down the road to the small block, thereby increasing the averaging the transportation costs over a larger effective harvest area. Thus, it may be possible to accept some harvest blocks that are smaller than usual, such that an average minimum block size may better represent operational reality.

Accounting for watercourse buffers

In this study, we made the assumption that watercourse and wildlife habitat buffers had restrictions on harvesting. Although the Clean Water Act does allow for certain harvest operations within buffers, the participating Licensees did not take advantage of them, presumably because the volumes extractable were not justifiable in terms of cost. Thus, the watercourse buffers were very real barriers to harvesting.

Licensees who do not have GIS facilities rely on DNRE to provide them with hard copy base maps and data files indicating the area of various stand types in the unrestricted land base. In general, the areas of forest classes used in the baseline models were determined by summing the area of operable stands less the area within buffer zones. Thus the stratified database corresponds to the available area on the ground. However, in disaggregating the classes to find individual stands, the linkage between forest classes eligible for harvesting and actual stands on the ground uses the base Forest Development Survey (FDS) map and stand numbers prior to blocking. Had we proceeded in this manner as well, the *Crystal* algorithm would first generate blocks without considering buffers. Afterwards, we would have had to overlay the buffers over the blocks.

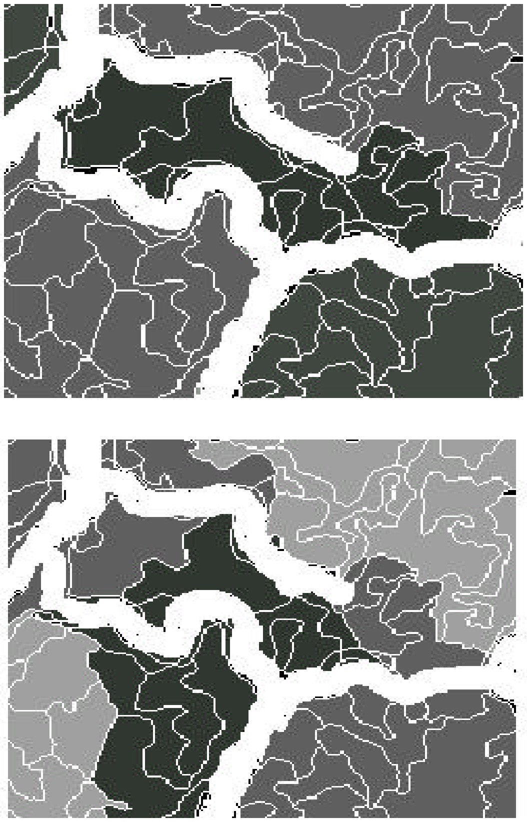


Figure 21. Implications of blocking after buffer overlays applied (top) and before watercourse buffers applied (bottom). Stands shaded similarly represent a single block.

Figure 21 depicts 2 hypothetical *Crystal* allocations. In the upper illustration, watercourse boundaries were in place before blocking whereas in the lower illustration, buffers were overlaid after blocking. Notice that the blocks in the bottom illustration are subdivided by the buffers resulting not only in reductions in effective size, but also necessitating additional stream crossings.

Whether the buffers are taken into account before or after the blocking process has a great impact on the resulting blocks themselves and on scheduling. If buffers were applied after blocking, it is conceivable that a block could be subdivided into 2 or more parts by a watercourse buffer running through it. Even if the block were not subdivided, the restrictions on stream crossings may require very long skidding or hauling distances for machinery operating on the block. Conversely, stream buffers act as barriers between adjacent stands, and if blocking is done after the buffers have been applied (as was done in this study), it may be difficult to find contiguous areas large to enough to produce a feasible harvest block.

Block harvest scheduling issues

Planning Horizon

Comparing results of the *Block* harvest schedules to those of the Licensees is difficult. The blocks generated by *Crystal* were composed only of stands belonging to development types scheduled for harvest in the tactical planning horizon; we do not know the criteria used by the Licensees during blocking, but we assume that they did not limit themselves only to the forest area scheduled by FORMAN+1 in the tactical planning horizon. Thus it becomes difficult to directly compare block layouts.

Although we discussed the matter with each of the Licensees and with DNRE staff, it was not clear to us what the exact requirements were for blocking. In general, it appeared to us that Licensees were to lay out harvest blocks for all planning periods up until regeneration stand types were queued for harvest. By restricting blocking only to currently existing stand types, both Licensees could block out the harvests arising in the first 7 planning periods. Once these blocks were determined, we were not sure how many of these blocks were actually scheduled for harvest; the forest management manual requires a 25 year plan but no details are provided on how the block harvest scheduling is to be performed.

One concern we had was that by blocking out a harvest schedule by hand, the intent of the strategic plan could be compromised inadvertently. Depending on how much area in the blocks belongs to classes scheduled for harvest outside the tactical planning horizon, there may be severe deviations from the intended strategic plan. Recognizing that there will always be some area that simply cannot be economically extracted, we limited the blocking process only to stands eligible for harvest in the first seven periods. Thus, despite the fact that *Block* was permitted to assign most blocks to any period within the planning horizon, none of the blocks actually scheduled for harvest include development types scheduled for harvest in the post-tactical planning periods.

In addition to restricting the area for blocking only to areas scheduled for harvest during the tactical planning horizon, we also believe that the planning horizon for scheduling should include the same number of periods as was used to develop blocks. For this study, this means that the planning horizon for the *Block* runs was 7 periods, even though only the initial 5 period schedule would be used as the basis for a management plan. Otherwise, the possibility exists for inadvertently overestimating the allowable cut.

For example, if one were to block out 7 periods of harvests but only 5 periods were used in the planning horizon. two periods worth of blocks would be available to you to work around opening size and adjacency delays. Even if *Block* did not over-harvest (although it is also quite possible), the unharvested blocks could be configured such that significant adjacency conflicts arise later on.

Verifying sustainability

Once a block harvest schedule is developed (using any procedure) it is typically tested by running the block schedule through the strategic planning model to guarantee that the harvest level from the block schedule is sustainable. What remains unclear to us is how the sustainability criterion is measured. If we consider a pre-block AAC of 100 000 m3 for the strategic planning horizon and a post-block AAC of 90 000 m3 for the tactical planning horizon, what is the required AAC for the period after the tactical planning horizon?

Given the procedure we used, we suggest that the AAC should rebound to close to the original pre-block AAC, because we considered for blocking only development types scheduled for harvest in the tactical planning horizon - those in the periods following the tactical planning horizon are still available, and except for changes in the timing of regeneration on harvested blocks, the assumptions used in the original strategic model should still hold. Considering that in most cases, less than the full amount of area scheduled for harvest during the tactical planning horizon will be harvested, there should be surplus area available for harvest in the post-tactical planning periods. Thus, if the AAC for the post-tactical planning horizon does not rebound nearly the pre-blocked AAC, we suggest that the block harvest schedule has somehow impaired the productivity of the forest.

Although we do not have a definitive answer, a question which needs to be raised is, what factors determine whether a block pattern and harvest schedule is good or not? Obviously, some desirable criteria include sustainability of harvest flows into the future, consistency in terms of harvest costs and operating procedures, flexibility sufficient to not preclude future options, and the ability to create these patterns and schedules quickly. However, different agencies, and indeed different individuals, will ascribe various degrees of importance to these criteria; there is no way to favor one without concessions from the others.

Limitations of Block

Of the three models used in the study, *Block* is the oldest and the most limited in usability. *Block* was designed to be a user-interactive blocking tool, and at the time of its development, clearcut harvesting was largely the only harvesting method employed in the province. No automated blocking tools existed and so a single input file which listed all relevant information about blocks on a single line was not inappropriate. Despite being a research tool it was possible to use *Block* on a real sized problem, however, we did identify several limitations in the algorithm.

Single entry harvests

Like *Crystal*, *Block* is limited to single entry harvests. However, with *Crystal*, a multiple entry harvest still requires only one harvest block to be established. With *Block*, yields occur at several points during the planning horizon, and so it must be able to keep track of each harvest entry, to ensure that they occur in the proper sequence.

Adjacency

Although the input file structure is easy to understand, it has a number of deficiencies. *Block* input files are ASCII text files which are limited to 255 characters in length. Because all information about a block is stored on a single line, keeping track of multiple products or increasing the number of adjacent blocks results in longer line lengths. With three products, *Block* is limited to a maximum of about 12 adjacencies per block. We encountered numerous occasions where there were 15 block adjacencies and so we had to try a different block layout.

Volume flows

Block does not allow volume flows to be controlled on a periodic basis – the user can only specify minimum and maximum limits for products which apply in every planning period. In the case of License 8, where *Woodstock* projected an increase in the allowable cut in period 6, we had to assume a strict even flow of wood for the block harvest schedules.

Monte Carlo algorithm

The Monte Carlo integer programming (MCIP) algorithm used by *Block* is relatively simple but somewhat inefficient - it throws away a solution as soon as it detects an infeasibility. Although the algorithm can generate alternatives quickly, it may require a large number of attempts to find a feasible solution; this is particularly true when the problem is highly constrained. Moreover, the only way to know if a problem is feasible is when *Block* finds a feasible solution; if *Block* makes 10,000 unsuccessful attempts, the problem may be infeasible, or it may find a feasible solution on the next attempt.

Conclusions

The combination of *Woodstock*, *Crystal* and *Block* as an integrated system yielded substantial savings in time and effort to produce a spatially feasible block harvest schedule. Once the linkages and conversion routines were in place, the time required to prepare the GIS database, produce a strategic harvest schedule from the baseline data sets, allocate the first seven periods of that schedule to blocks and finally schedule the harvesting of those blocks for seven periods was approximately 36 hours of labour and an additional 49 hours for computer processing. The maximum time required for individual tasks for a given License is given in the following table.

Item	**Labor**	**Batch processing**
Woodstock		
Develop classification scheme	8 h	1 hr
Set up overlays	1 h	19 h

Convert models from FORMAN+1 to *Woodstock*	8 h	
LP set up	8 h	
Solving		5 h
Crystal		
Set up input files	4 h	
Allocate blocks		22 h
Block		
Set up input files	3 h	
Process 4 block configurations	4 h	3 h
TOTAL time	**36 h**	**49 h**

Woodstock

We found that linear programming has significant advantages over the inventory projection method used in FORMAN+1. Despite our unfamiliarity with the Licenses and the baseline models provided by the Licensees, we were able to produce *Woodstock* equivalents relatively easily. Furthermore, on the basis of constraint capabilities, output levels and effort required to find an acceptable solution, the *Woodstock* models clearly outperformed their baseline counterparts. Based simply on AAC, the *Woodstock* models yielded a 9% increase on License 8 and a 22% increase on License 4. Furthermore, MCFH requirements were met in all planning periods on both Licenses.

Crystal

Crystal was shown to work well despite a few shortcomings. The inability to block out more than a single harvest prescription is a potential problem although it was not an overwhelming limitation in this study. On the other hand, compared to the manual approaches used by the Licensees, *Crystal* is a vast improvement. It does not require a particular harvest scheduling model; we used *Woodstock* in this study, but FORMAN+1 or any other stratum-based planning model could have been used to determine the harvest schedule. Many alternative block configurations can be generated by *Crystal* in a single day, and since it is possible to accommodate previously determined blocks, there is ample opportunity user intervention into the blocking process. Most important, unlike the manual process which is subject to human error, *Crystal* provides a consistent, reproducible method of generating harvest blocks

The allocation algorithm used in *Crystal* was sensitive to the blocking parameters specified (particularly minimum block size) and to the nature of the land base on which it was working. Although we used the same sets of parameters on both Licenses, for a given set of parameters *Crystal* always allocated a higher proportion of scheduled area for License 4 than it did for License 8.

Block

Block performed adequately but required a significant amount of effort and custom programming to be able to use it efficiently. Despite its awkward input file structure and the inability to easily accommodate non-clearcut harvest prescriptions, *Block* produced good results in this study.

The assumption that smaller blocks would yield fewer conflicts for a given land base than a larger one (Clements et al., 1991) was borne out in this study: when the minimum block size increased from 10 to 20 ha, the number of unharvested blocks more than doubled.

One of our assumptions about *Block* did not materialize, however. We assumed that *Block* would group together blocks eligible for harvest in the same period, thus avoiding adjacency conflicts. However, it appears that the *Block* algorithm tends to deviate timing of harvest to avoid adjacency conflicts with only incidental grouping of compatible blocks.

Recommendations

Conduct a benchmarking exercise for the two Licenses in this study using the approach taken by the FORMAN 2000 group

A comparison of manual and computer-automated approaches to a complex task like forest planning is useful for indicating potential advantages and disadvantages but as a means of evaluating the automated system, it is somewhat limited. Algorithms can approximate the processes undertaken by an analyst, but they cannot reproduce insight, nor can they adapt to changing perceptions of a problem as a human mind can. In essence, the comparison becomes one of apples and oranges.

On the other hand, the FORMAN 2000 project was undertaken to do the same tasks that were undertaken in this study. Although the solution techniques used by the FORMAN 2000 models are significantly different than one presented in this report, they are both based on algorithms and thus comparisons can be drawn fairly.

Provide Licensees with detailed attribute and topological data from the provincial geographic information database

Geo-referenced forest planning models facilitate the design and development of block harvest strategies. However, GIS facilities are not required to undertake these activities but detailed attribute data and adjacency information are required. DNRE should consider providing Licensees without GIS facilities the following data:

- a master polygon attribute table for each License incorporating stand level information (including age estimates), site classification, watercourse buffers and wildlife exclusion zones
- a polygon adjacency table corresponding to the master PAT file which details stand adjacencies within and across map boundaries.

With this information, any Licensee could implement the strategy used in this study, or another one based on alternative models (O'Hara *et al.* 1989; Nelson *et al.* 1988).

Establish consistent guidelines for planning procedures and articulate requirements and regulations in terms that are not dependent on a particular modeling system

The number of periods that must be blocked out and scheduled should be specified in quantifiable terms (e.g., a specific number or all planning periods where harvesting occurs only in existing stand types). Output requirements (like mature conifer furbearer habitat) should not be based on a particular modeling approach since the assumptions underlying a particular model may not hold if another modeling approach is used. The use of jargon which is open to interpretation should be avoided, or if unavoidable, should be strictly defined in measurable terms – for example, how is the sustainability of a block harvest schedule measured?

Implement joint planning activities for Crown and Licensee freehold lands

Licensees with significant land holdings should consider conducting management planning exercises for Crown and company freehold lands simultaneously. First, there will likely be some allowable cut effect exhibited by combining land bases (Davis and Johnson, 1987). Second, combining the two land bases into one for the purposes of blocking would reduce the fragmentation problem and the ability to generate larger harvest blocks.

Implement linear programming techniques as part of a broad-based planning methodology

Linear programming has been used extensively in other jurisdictions to address complex forest planning problems like those arising in New Brunswick, and as such, numerous modeling systems based on LP have been developed. Furthermore, expertise gained by Licensees in the area of LP-based forest management scheduling can be applied to other planning problems such as equipment deployment in operations planning.

Implement modifications to Crystal

Control the blocking process by setting a minimum average block size rather than a fixed minimum. This would allow for some small blocks where necessary by countering them with a few larger blocks as well.

Change the blocking algorithm to progressively accept smaller blocks until the average block size reaches the minimum allowed. This would reduce the number of small blocks produced and yield block layouts which better reflect long-term objectives by minimizing timing choice deviations.

Implement modifications to Block

Restructure input and output files as dBASE files. This would circumvent *Block*'s current limitations on problem size and facilitate links to *Crystal* and GIS data.

Modify the MCIP algorithm to attempt changes to infeasible solutions before discarding them. A more efficient algorithm would increase the probability of finding better overall solutions.

Allow the user to specify harvest flow constraints for individual planning periods. This would allow for planned increases or decreases in the harvest of one or more products in designated periods.

Literature cited

Clements, S.E., Dallain, P.L., and Jamnick, M.S. 1990. An operational, spatially constrained harvest scheduling model. Can. J. For. Res. 20: 1438-1447.

Dallain, P.L. 1989. A block harvest scheduling model for spatial management planning. M.Sc.F. thesis, University of New Brunswick, Fredericton.

Davis, L.S. and Johnson, K.N. 1987. Forest management. McGraw-Hill Book Co., Toronto.

Jamnick, M.S. 1990. A comparison of FORMAN and linear programming approaches to timber harvest scheduling. Can. J. For. Res. 20: 1351-1360.

Nelson, J., Brodie, J.D. and Sessions, J. 1988. Integrating short-term spatially feasible harvest plans with long-term harvest schedules using Monte-Carlo integer programming and linear programming. In Proceedings of the 1988 Symposium on Systems Analysis in Forest Resources, 29 March - 1 April, 1988. USDA For. Serv. Rocky Mountain. For. Range Exp. Stn. Gen. Tech. Rep. RM-161. pp 224-229.

O'Hara, A.J., Faaland, B.H. and Bare, B.B. 1989. Spatially constrained timber harvest scheduling. Can. J. For. Res. 19: 715-724.

Walters, K.R. 1991. Spatial and temporal allocation of strata-based harvest schedules. M.Sc.F. thesis, University of New Brunswick, Fredericton.

Appendix

An overview of linear programming models

When *Woodstock* is used to analyse LP models, the program operates in two different modes: matrix generation mode and report generation mode. During matrix generation, the *Woodstock* interpreter uses the built-in simulator to enumerate all possible outcomes for every silvicultural action in every planning period. For example, consider a stand which may be clearcut either in the first planning period or in the second. Furthermore, the clearcut stand may be either naturally regenerated or planted to black spruce immediately following harvesting. If we enumerate the possible choices for this stand (ignoring any future harvesting for the moment), we quickly see that there are four alternatives for this stand:

1. clearcut and naturally regenerate the stand in period 1
2. clearcut and naturally regenerate the stand in period 2
3. clearcut and plant the stand to black spruce in period 1
4. clearcut and plant the stand to black spruce in period 2

If we consider that the regenerated stand (naturally regenerated or planted) could be clearcut harvested and naturally planted when it is either 8 or 9 periods old, we double the number of possible outcomes (decision variables) to consider for this stand:

1. nat. regen. in period 1, clearcut/nat. regen. in period 9
2. nat. regen. in period 1, clearcut/nat. regen. in period 10
3. nat. regen. in period 2, clearcut/nat. regen. in period 10
4. nat. regen. in period 2, clearcut/nat. regen. in period 11
5. plant BS in period 1, clearcut/nat. regen. in period 9
6. plant BS in period 1, clearcut/nat. regen. in period 10
7. plant BS in period 2, clearcut/nat. regen. in period 10
8. plant BS in period 2, clearcut/nat. regen. in period 11

When we consider that there are often many more silvicultural options for a stand than just two, and that there are many more timing choices for harvesting than just two, the number of decision variables for even a few stands or stand types becomes rapidly large. For a typical License, the number of alternatives to consider can be in the hundreds of thousands, far too many to evaluate using a simulation approach.

Once all the alternatives are evaluated, the matrix generator sets up the problem in a special form called a linear programming matrix (hence its name). A LP matrix is composed of an objective function (the quantity to be maximized or minimized) and a series of constraints which limit the availability of various resources. For example, if we have 8 different choices for an 80 ha piece of land, the total area assigned to all of the choices cannot exceed 80 ha. In other words, a land area constraint must be imposed to bound the problem: $x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8 = 80$. Any one of the decision variables may take on values from 0 to 80, but the sum of all variables must equal 80. Other constraints are used to control the amount of timber produced each period, the amount of planting done each period, et cetera.

Once the LP matrix has been generated, an optimal solution to the system of equations represented in the matrix is found. After processing, the LP solver may report one of three conditions: infeasible, unbounded or optimal. An infeasible problem is one where one or more constraints cannot be satisfied. For example, if a constraint requires that 10,000 m3 be produced but only 8,000 is possible, then the problem is infeasible as formulated. An unbounded condition indicates an error in the matrix and typically will not occur under normal conditions. An optimal solution represents a combination of activities that produces the highest (maximization) objective function value; more than one optimal solution may exist, but any such solution will produce the same objective function value.

Once an optimal solution is found, the combination of activities represented by it needs to be communicated to the simulation routines in *Woodstock*; this is accomplished through a translation utility which reads the solution file generated by the LP solver and writes out an action sequence file. The sequence file replaces the Queue section and selection rules of a regular simulation model by specifically telling the simulator what to do and when to do it. However, in all other respects, *Woodstock* functions as it normally does, providing run-time graphics and reporting capabilities. The function is analogous to report writer programs of other LP based harvest scheduling models.

We formulated the management problems of the two participating Licensees in the form of what is commonly known as a generalized model II LP. Without getting into a lot of detail about different types of LP models, this form of LP allows you to have in your models multiple outcomes following silvicultural treatments, explicit mortality due to age and regeneration to ages other than zero. Although it is a simulation model, FORMAN+1 essentially uses the same Model II structure, despite allowing multiple outcomes only for clearcut harvests. Since our analysis was based on FORMAN+1 models, the generalized Model II LP form was appropriate.

Most references in the prevailing forest management literature suggest a planning horizon of 2 -2+ rotations and given New Brunswick conditions, we felt that 100 year planning horizons were more appropriate than the minimum 80 year planning horizons required by the Province. The rationale for longer planning horizons is to estimate wood supply beyond the time where the current forest structure ceases to influence AAC. Although even a 100 year planning horizon does not extend so far into the future, it seemed to be a reasonable compromise.

An overview of the Crystal algorithm

Crystal uses three types of information to allocate a harvest schedule to stands: (1) the schedule of activities by development type, planning period and area to be treated reported by *Woodstock*, (2) a list of adjacencies detailing pairs of stands that are considered to be neighbours extracted from the ArcInfo data files, and (3) a list of stands which details the area of each stand, its age and the development type it belongs to for yield prediction purposes (based on classification scheme used in *Woodstock* model and ArcInfo polygon attribute files). These three lists are stored in relational database files, linked by two key fields: stand/map number and analysis area ID number.

Crystal attempts to allocate stands to harvest units by grouping together stands scheduled for harvest in the same period, continuing to add stands to the unit until the user-specified target size is reached. Once the target size is reached, a new starting point is selected and *Crystal* attempts to generate another harvest unit. In some cases, it may be desirable to allow timing choice deviations. For example, one stand may be scheduled for period two while all others around it are scheduled for period one. Rather than leave an isolated stand unharvested, *Crystal* can include the stand in the harvest unit by accepting a one-period timing choice deviation. On the other hand, a similar stand may actually have other neighbours which are also scheduled for period two. In this case, it may be preferable not to accept the timing choice deviation since the period two stands may comprise a feasible harvest unit on their own. User's can control how *Crystal* deals with these situations, by specifying the periodic deviations allowed for Type 1 stands (the first case) and for Type 2 stands (the second case).

Finally, after all possible allocations have been made according to the user specified rules, *Crystal* can invoke a cleanup routine. This routine scans the eligible stand list for unallocated stands and attempts to incorporate as many of them as possible into previously allocated harvest units. Many of these orphan stands are generated because *Crystal* simply stops allocating stands when the target size is reached. Others however simply cannot be allocated using the user-specified restrictions on timing choice deviations. Rather than leave these stands unallocated (and thus unharvestable), it may be preferable to allow larger than normal timing choice deviations to accommodate them. Again, the user specifies the periodic deviations allowed.

With virtually any planning model results for the final planning period may need special consideration. Because the forest system within the model ceases following the final planning period, the final planning period often exhibits unexpected or at least counter-intuitive behavior. For example, harvest levels may change radically in the last period even though the harvest had been constant for all previous periods; or silvicultural activity may simply cease near the end of the planning horizon. Generally, such behaviour is attributable to model artifacts: an arbitrary stopping point in an otherwise continuous system. For this reason, it is usually wise to extend the planning horizon by one or two periods to ensure that such behaviour does not inadvertently affect the planning interval of interest. Given that the Forest Management Manual requires a 5 period planning horizon, we chose to allocate the first 7 periods of the *Woodstock* schedule so that the allocation algorithm used in *Crystal* would be able to accelerate or delay harvesting blocks in period 5 as easily as earlier periods.

An overview of the Block scheduling algorithm

The *Block* program uses a single input file to store nearly all information required for block harvest scheduling. Each harvest block is identified by a unique name or number, its size is given, the management unit and habitat zones each block belongs to is given, a list of all other blocks adjacent to each block is listed, as well as periodic volumes for specific harvest products harvested from each block. The input file also specifies the length of the planning horizon, the number and names of management units and habitat zones, the names of products, and the maximum number of adjacent blocks. In addition, the user may provide a list of blocks which can be harvested only in certain periods; any blocks not appearing in the block availability list are considered available for harvest in all periods. Finally, the user provides at run-time the objective function and harvest flow constraints which are the basis for determining a feasible solution.

Because of the number of alternatives that exist in a block harvest scheduling problem, and the number of constraints that would be required to maintain adjacency delays and opening size requirements, a deterministic optimization approach such as integer programming is not workable. Instead, *Block* uses a Monte Carlo integer programming (MCIP) algorithm to find feasible block harvest schedules.

The MCIP-based random harvest procedure begins searching for a feasible solution by assigning all candidate harvest blocks to random positions in a harvest queue. A random number between 0 and 1 is generated for each block and the blocks are sorted in ascending order. Starting at the beginning of the queue, blocks that are available for harvest in the current period are successively harvested. *Block*s adjacent to harvested blocks are prevented from being harvested for the specified exclusion period if the maximum opening size for the neighborhood has been reached. The harvest procedure continues until the minimum volume requirements have been met, or until all available blocks in the queue have been exhausted. If the minimum volume requirements are met, harvesting begins for the next period. Otherwise, the solution is deemed infeasible and the current attempt at finding a feasible solution is aborted.

For volume maximization problems, a continue-harvest procedure attempts to improve a basic feasible solution by harvesting as many additional blocks as possible without violating any of the adjacency, block availability or maximum harvest flow constraints. This is accomplished by testing each block in each period to see that none of the constraints are violated before harvesting. *Block*s which violate one or more constraints in any period remain unharvested.