Model I Linear Programming Versus Integer Metropolis Solutions

Paul C. Van Deusen and Jonathan Aggett*

Draft: April 7, 2004

Abstract

An implementation of the Metropolis algorithm for solving forest harvest scheduling problems is compared to analogous linear programming solutions. The first step is to formulate the problem for each algorithm in a comparable way and without spatial constraints. A model I formulation is used in order to track each management unit. The second step is to look at the impact of spatial constraints that can be handled by the Metropolis Algorithm, but not with linear programming. The reduction in output due to spatial constraints is shown to be problem dependent and potentially large.

*NCASI, 600 Suffolk St., Lowell, MA. 01854, PVandeusen@ncasi.org

Contents

1 Introduction

Forest harvest scheduling problems are often solved with linear programming (LP) based on Model I or Model II formulations. This categorization is often attributed to (Johnson and Scheurmann, 1977) and refers to the nature of the LP decision variables. Model I decision variables are acres in a management unit, while Model II decision variables are acres in a management class, e.g. an age-class. Model II formulations are useful for reducing the number of decision variables, but make it more difficult to handle spatial issues or track individual acres.

One of the objectives of this paper is to look at the effect of going from a continuous LP solution to an integer solution where only 1 management regime can be assigned to each management unit. The Model I formulation is the focus of this paper, because of its inherent ability to track individual management units. One of the first published Model I formulations was Curtis (1962), so the basic LP formulation used here is very well known and tested. An LP solution is optimal for the given problem formulation, which is an important attribute. However, there are often non-linear problem constraints that can't be included in the LP formulation. For example, Model I formulations should usually be constrained to have integer solutions, which is beyond the capabilities of commercial LP solution algorithms. Thus, the optimal LP solution can not be implemented when only 1 management regime is allowed per management unit. Furthermore, integer programming algorithms are not able to handle operational size harvest scheduling problems.

There are heuristic methods that can find integer solutions for operational size problems, although they may not guarantee optimality. Tabu search (Bettinger et al., 1997; Caro et al., 2002), the genetic algorithm (Mullen and Butler, 1997) and the Metropolis algorithm (Metropolis et al., 1953) are 3 well known examples. Simulated annealing (Lockwood and Moore, 1993) is a special case of the Metropolis algorithm that has been used for forest harvest scheduling. A comparison is made here between the Metropolis algorithm as used by Van Deusen (1999, 2001) and a Model I LP formulation. The Metropolis algorithm provides integer solutions and otherwise imposes constraints that are similar to the LP constraints. However, the Metropolis formulation used here is not identical to the LP formulation and doesn't guarantee optimal solutions.

Heuristic algorithms are used primarily because they are able to handle new issues that LP is unable to address. These new issues often involve spatial considerations, such as limiting clearcut block size. For example, the major forest-industry landowners in the USA have agreed to abide by a set of sustainable forestry initiatives (SFI™). SFI requires landowners to control clearcut size and improve wildlife habitat. Companies that don't follow SFI guidelines can not be memebers of the American Forest and Paper Association (AF&PA 1994).

2 Review of Algorithms

Harvest scheduling, as defined here, requires a land area broken into polygons with a list of potential management regimes assigned to each polygon. A management regime defines a set of activities that can occur over the planning horizon. Specifically, a regime gives year-output pairs to denote the timing and amount of output that will occur if this regime is followed. A schedule is produced by assigning one regime to each polygon. However, the combinatorial magnitude of the harvest scheduling problem results in an enormous number of potential schedules. If there are P polygons with each polygon having R possible regimes, then there are R^P possible schedules. This makes it clear that even an extremely small problem, say R=5 and P=50, generates very many schedules.

It follows from the above discussion that a harvest scheduling algorithm assigns regimes to polygons to create a management schedule. A regime specifies a list of years where an output occurs. The output could be a volume of wood, a cost, a present net value, or acres of habitat. Optimality could be based on maximizing present net value (PNV), maximizing habitat diversity, or another user specified criterion. Given the huge number of potential schedules, it is likely that sub-optimal schedules exist that might achieve over 99% of the optimal objective function value. This provides the justification for looking at alternatives to LP that handle important non-linear constraints and provide near optimal solutions.

2.1 Metropolis algorithm

The Metropolis formulation used here has been described elsewhere (Van Deusen, 1999, 2001). The first step is to select an initial schedule, possibly at random, which evolves at each iteration. The schedule at iteration r is represented by the vector $X^r = x_1^r, ..., x_N^r$, where x_i^r represents the regime assigned to polygon i at iteration r. Each iteration of the

algorithm creates a new and improved schedule that can be compared with other schedules using an objective function $E(X^r)$.

The overall objective function consists of a weighted sum of components, where each component represents a particular sub-objective. The value of the objective function at iteration r is

$$E(X^r) \;=\; \sum_{j=1}^{J} w_j^{r-1} C_j(X^r) \tag{1}$$

where w_j^r is a weight determined from the iteration r schedule, and $C_j(X^r)$ is the jth objective function component evaluated at the rth schedule.

The Metropolis algorithm iteratively considers new proposal regimes as substitutes for the current regime of each polygon. The proposal regime is immediately accepted if it improves the overall schedule by decreasing the objective function (1). Proposals that are worse than the current regime can still be accepted according to a computed probability. It is possible to escape a solution region that represents a local minima by sometimes accepting changes that don't improve the objective function. This algorithm does not attempt to converge on a single best solution by slowly adjusting a so-called temperature parameter as with SA as presented by (Lockwood and Moore, 1993). Rather, the individual component weights can be adjusted for finer control of convergence properties.

The objective function is a sum of components multiplied by weights. The weights, w_j^r , are critical to the performance of the algorithm, since they determine the relative importance of each objective function component. There is no way to know in advance what the correct value for a weight should be, since this will vary according to the data and the combination of components in the objective function. The user determines the weights indirectly by defining lower and upper goal limits between 0 and 1 that specify the level of attainment desired. Attainment is quantified by evaluating a goal function at the end of each iteration. The component j goal function is $g_j(X^r)$, which depends on the current schedule. Weights are adjusted after each iteration as follows:

- if $g_j(X^{r-1}) > U_j$ then $w_j^r = a \cdot w_j^{r-1}$,
- if $g_j(X^{r-1}) < L_j$ then $w_j^r = w_j^{r-1}/a$.

where U and L are the user specified upper and lower limits and a is an adjustment factor between 0 and 1. The algorithm is said to have converged for a particular goal when $g_j(X^r)$

is between U_j and L_j. Otherwise, the weights are decreased when the goal is over-attained and increased when it is under-attained. This use of goal functions (Van Deusen, 1999, 2001) makes it possible to write a computer program that is capable of solving harvest scheduling problems with quite general objective functions.

The suggested algorithm is:

1. Choose J objective function components as in equation (1) along with a goal function adjustment parameter $0 \leq a \leq 1$.
2. Initialize X^1 by choosing a regime for each polygon (possibly at random), and let $w_j^0 = 1$, for $j = 1, \ldots, J$. Set r=0.
3. Set r=r+1 and for polygon i=1,...,N:
   a) Perturb X^r into Z by choosing a new regime, k, at random for polygon i.
   b) Let $p^* = min\{1, exp[E(X^r) - E(Z)]\}$
   c) Accept regime k for polygon i with probability p^*.
4. Evaluate the goal functions, g_j for j=1,...,J and adjust the corresponding weights if necessary.
5. Repeat (3) and (4) until the weights have converged, or the problem is declared infeasible.

2.2 Linear programming

The linear programming algorithm is well known and is based on maximizing or minimizing an objective function subject to constraints. It is generally specified (Leuschner, 1990) as

$$\begin{aligned} Max \quad \boldsymbol{Z} &= \boldsymbol{cx} \\ subject\ to \quad \boldsymbol{Ax} &\leq \boldsymbol{r} \\ \boldsymbol{x} &\geq \boldsymbol{0} \end{aligned}$$

where **Z** is the objective function value, **c** is a vector of coeffcients representing the contribution of the decision variables, **x** is a vector of decision variables, **A** is a matrix of

coefficients measuring the effect of the constraints on the decision variables, and **r** is a vector of constraints.

The x-variables for linear programming are related to the x-variables in the Metropolis formulation in that both algorithms define the schedule by assigning values to the x-variables. The Metropolis algorithm has one x-variable for each polygon and assigns a single regime to each x. The LP Model I formulation requires one x-variable for each combination of polygon and regime. The LP solution algorithm assigns non-zero values to the x-variables that enter into the optimal solution, and can assign multiple regimes to one polygon. Constraints must be defined to keep LP from assigning more than the whole polygon to various management regimes.

A relatively simple LP formulation is used with the intention of being as similar as possible to the Metropolis formulation. This is based on decision variables, x, that represent the proportion of a polygon assigned to a particular regime so that $0 \leq x_{ij} \leq 1$ where subscript i is for polygon i and j denotes regime j. The objective function cost coefficients, c_{ij}, represent the total amount of output that results from assigning all of polygon i to regime j. For convenience, summary/accounting variables are appended to the x vector to keep track of acres cut and flow of goods by year. The cost coefficients are set to 0 for the summary variables.

The first set of constraints is imposed to keep LP from assigning more than 100% of the polygon to its set of valid regimes as follows

$$\sum_{j=1}^{J_i} x_{ij} \leq 1 \qquad i = 1, \ldots, n \tag{2}$$

where J_i is the number of valid regimes for polygon i.

Summary constraints are included to allow LP to keep track of annual output of items such as clearcut acres and wood flow. Call the summary variables s_{kt}, where k indicates the kind of summary variable and t indicates the year. The summary constraints have the form

$$\sum_{x_{ij} \in k,t} a_{k,j} x_{i,j} \; - \; s_{k,t} \; = \; 0 \qquad \forall \; k, t \tag{3}$$

where the relevant x_{ij} are those polygons and regimes that contribute to the output that is being accounted for in year t, and the a_{kj} coefficients give the contribution of each polygon to summary variable $s_{k,t}$.

Finally, flow constraints are imposed to control the trend and variability over time of the

summary variables. A typical set of flow constraints takes the form

$$\begin{aligned} s_{k,t+1} &\leq (1+U)s_{k,t} \qquad t = 1,\ldots,T-1 \\ (1-L)s_{k,t} &\leq s_{k,t+1} \qquad t = 1,\ldots,T-1 \end{aligned} \tag{4}$$

where U and L are user defined upper and lower proportions to control the change in flow from 1 time to the next.

3 Example Applications

Comparisons are made between the Metropolis and LP algorithms using realistic harvest scheduling data. The first example uses a small dataset with 419 planted pine stands. The second application uses a somewhat more complicated dataset with a mixture of pine and hardwood stands. A third example uses a larger dataset with 1046 polygons from the southern US. Included in this dataset are: natural hardwood stands, natural pine stands, planted pine stands, seeded pine stands, and various non-forested sites. The examples show how close one can expect to come to the optimal LP solution with the Metropolis algorithm. They also demonstrate that the cost of spatial constraints are problem dependent.

3.1 Example Application 1

The first example uses a simple harvest scheduling data set consisting of 419 loblolly pine stands from the southeastern US. Management regimes 1 through 15 are to clearcut the stand in years 1 through 15 and regime 16 signifies no cutting during the 15 year planning period. The flow of wood in total dry tons will be controlled along with the number of clearcut acres (1 hectare=2.47 acres) each year. Also available are the present net value associated with each regime for each stand. The PNV from regime 16 is arbitrarily set to 1, which is much less than for any other regime. Not every stand has all 16 regimes potentially available. If every stand is assigned to the highest allowed PNV regime, the total PNV is 552,882.

3.1.1 Linear Programming

An LP run is made where the objective is to maximize PNV subject to flow constraints on tons and clearcut acres, where U and L in equations (4) are set to 0.1. These constraints reduce the objective function value to 548,673. The trends in clearcut acres and wood output over time are relatively smooth (Fig 1). A summary of regime assignments (Table 1) shows that the optimal solution was non-integer. The x-column (Table 1) represents the mean of the decision variables for each regime. There are 419 polygons in this dataset, but there are 442 non-zero decision variables. This indicates that 23 polygons had multiple regime assignments. The do-nothing regime, 16, was assigned as the only regime for 37 stands, whereas clearcut in year 1 was assigned at least partly to 37 stands. To meet evenflow requirements, every clearcut regime had to be assigned, since regimes 1 through 15 correspond to cutting in years 1 through 15.

3.1.2 Metropolis Algorithm

Now we compare the Metropolis algorithm's integer solution with the LP result. This involves 2 steps. First, we look at the Metropolis solution derived independently of the LP solution. By definition, this must attain less PNV than the optimal LP solution. Second, we message the LP solution to be integer by assigning each polygon to its principal regime, i.e. the regime that is assigned the largest proportion of the stand.

A Metropolis solution was obtained using an objective function that was as similar as possible to the LP formulation. The PNV was 535,164 which is 2.5% less than the LP result. The trends in clearcut acres and wood output (Fig 2) are similar to the LP result, but somewhat less smooth. To facilitate direct comparison, the LP solution is forced to an integer solution reducing the PNV to 548,637, which is 1% less than the optimal solution.

3.1.3 Block Size Distribution

The block size distribution is based on a 2 year green-up window. This means that a clearcut stand contributes to the local clearcut block for the next 2 years. Controlling block size distribution is important for maintaining compliance with SFI™. It is instructive to look at the block size distribution (Fig 3) that results from implementing the forced-integer LP solution. The spatially unconstrained solution is unacceptable, since there are a number of clearcut blocks that exceed 1000 acres.

The Metropolis algorithm is able to produce a spatially constrained solution with all blocksizes (Fig 4) being between 20 and 180 acres with a mean of 97. The unconstrained solution has blocks ranging from 3 to 1193 acres with a mean of 192.5. The spatially constrained solution (Fig 5) achieved 80% of the optimal PNV with flows that are somewhat reduced from the unconstrained result.

3.2 Example Application 2

This is an application to a problem with 483 polygons representing mostly planted pines and some natural hardwood. There are 281 pine polygons and 99 hardwood polygons. The remaining non-forest polygons are included in the data for the purpose of manipulating the spatial distribution, but they are ignored in an LP non-spatial solution. The growth data came from stand level growth and yield models. The planning horizon for this problem is also 15 years.

The objective function is set up to potentially control the flows of:

1. total cubic foot volume (CFV),
2. clearcut acres,
3. the acres in forested habitat,

The flow of forest habitat results from pine with at least 1000 cubic feet per acre. One pine stand could contribute to this habitat until year 10, when it is clearcut. Likewise, another stand of pines that is too small at year 1 might begin to add to forest habitat at year 10 if it is allowed to grow.

3.2.1 Linear Programming

The LP run constrains the cubic volume flow to be between 90 and 110% of the previous years value. The forested habitat flow is left unconstrained. This results in an objective function value of 1.35e+08 cubic feet, which is 82% of the unconstrained maximum (1.65e+08). In this case, the objective function is maximizing the number of cubic feet removed over the planning horizon.

The LP solution split a number of stands between more than one regime (Table 2). The clearcut regimes are preceded with a CC, and the thinning regimes are preceded with a T. Thinning regimes are applied to hardwood stands, and the second year in a "T"-regime denotes a second thinning. The trends over time for all three flows, whether constrained (volume removed) or not constrained (forested habitat and clearcut acreage), are relatively smooth (Fig 6).

3.2.2 Metropolis Algorithm

With an objective function value of 1.32e+08, the Metropolis solution to the problem attains about 79% of the maximum. The flows are similar to those from LP, but the clearcut acreage and annual volume flows are somewhat smoother than those from LP (Fig 7).

3.2.3 Block Size Distribution

The blocksize distribution (Fig 8) for the spatially unconstrained linear programming solution includes clearcut blocks up to 1500 acres over a 3 year green-up window. These large blocks put this schedule out of compliance with SFI™requirements.

A spatially constrained Metropolis solution was obtained where the maximum blocksize was 175 acres (Fig 9) and the average was somewhat less than 120 acres. The spatial constraint comes at considerable cost, since the objective function was reduced to 49% of the maximum, or 8.12e+07. In this case, all blocks were constrained to be between 10 and 175 acres in size, with a 3 year green-up window.

3.3 Example Application 3

This is an application to a somewhat more complex harvest scheduling problem than the two preceeding examples. The dataset for this problem contains 1046 polygons representing a variety of forested and non-forested sites (Table 3). The regime classes are similar to those in Example 2, but the planning horizon for this example is 20 years. The Planted and Seeded Pine will be assigned to clearcut regimes, and the Natural Hardwood and Pine will receive thinning regimes.

The objective function for this problem is set up to potenially control the flows of:

1. wood volume,
2. "Big Pine" habitat,
3. end of period age-class distribution and
4. clearcut acres

3.3.1 Linear Programming

Initially, an unconstrained LP run is made, where the objective is to maximize the wood volume removed (cubic feet) over the planning horizon. This maximum occurs when the maximum wood volume regime is assigned to each polygon, without considering any other factors. Thus, if every stand is assigned to the highest allowed wood volume regime, the total volume is 9.04e+07 (where volume is measured in cubic feet).

Subsequently, an LP run is made where: the cubic volume flow is constrained to be between 80 and 103% of the previous year's value, and the Big Pine habitat flow and end-of-period age-class distribution are left unconstrained. The resulting objective function value is 6.59e+07 cubic feet, which is approximately 72% of the unconstrained maximum.

As usual, the LP solution split a number of stands between more than one regime (Table 4). The regime classes for this example are similar to those for Example 2. Hence, clearcut regimes are preceded with a CC, and thinning regimes are preceeded with a T, with the second year in a "T"-regime denoting a second thinning. The "BARE" regime refers to the planting of bareland, and the subsequent clearcutting of plantations. Hence, the first number in the "BARE" regime refers to the year in which planting is done, and the second number refers to the year in which the plantation is clearcut.

The flows over time for volume removed are not particularly smooth (Fig 10), although they do suggest an increasing trend. This trend, together with the trend of decreasing acres of Big Pine habitat (Fig 10) suggests that harvesting may have been delayed somewhat due to a lack of mature stands in the beginning of the planning period. The end-of-period age-class flow does not display a particularly smooth trend (Fig 10). In keeping with the high volumes of wood removed in the latter five years of the planning period, it would appear that a lot of the stands only reached maturity, and were subsequently harvested, near the end of the planning period. This delayed harvest explains the low acreage in the older age classes. The annual clearcut acreage is not directly controlled, and is therefore relatively uneven (Fig 10).

3.3.2 Metropolis Algorithm

With an objective function value of 6.15e+07, the spatially unconstrained Metropolis solution attains approximately 68% of the maximum, which is 4% less than that which the LP solution attained. Similar to Example 2, the flows for volume removed and clearcut acres are similar to those from LP, but somewhat smoother(Fig 11). The Big Pine habitat flow is almost identical to that from LP, but the end-of-period age-class flow is significantly smoother.

3.3.3 Block Size Distribution

The block size distribution is based on a 3 year green-up window. In keeping with the low clearcut levels, the spatially unconstrained, forced-integer LP solution yields few clearcut blocks, most of which are smaller than 400 acres (Fig 12).

The Metropolis algorithm was set up to produce a spatially constrained solution, with all blocksizes being between 0 and 200 acres (Fig 13). This spatially constrained solution achieved 57% of the maximum volume, or 5.15e+07, thus reducing the objective function value by 11% from the spatially unconstrained Metropolis solution.

4 Conclusions

The Metropolis algorithm compares favorably with LP for spatially unconstrained problems. The differences between LP and Metropolis results are problem-dependent to some degree. However, for the example applications considered here, the Metropolis algorithm was able to obtain integer solutions that were all within 4% of the LP optimal solution.

The spatially unconstrained solution for most scheduling problems will almost certainly have undesirable characteristics. The examples here demonstrated that clearcut block sizes can be enormous in spatially unconstrained solutions. Unfortunately, it can be quite costly to constrain blocksizes. This spatial constraint resulted in a 30% reduction in volume removed for Example 2 an 18% reduction for Example 1, and an 11% reduction for Example 3.

The Metropolis algorithm provides a reliable method to deal with spatial constraints on industrial size scheduling problems. LP can not address spatial constraints as well as heuristic approaches. However, it is a good idea to obtain a non-spatial LP benchmark by

which to judge the spatial result.

References

Bettinger, P., J. Sessions, and K. Boston (1997). Using tabu search to schedule timber harvests subject to spatial wildlife goals for big game. *Ecological Modeling 94*, 111–123. 2

Caro, F., M. Constantino, I. Martina, and A. Weintraub (2002). The 2-opt tabu search procedure for the multiperiod forest harvesting problem with adjacency, greenup, old growth and even flow constraints. *Forest Science 49*(5), 738–751. 2

Curtis, F. (1962). Linear programming the management of a forest property. *Journal of Forestry 60*, 611–616. 2

Johnson, K. N. and H. L. . Scheurmann (1977). Techniques for prescribing optimal timber harvesting and investment under dirrerent objectives – discussion and synthesis. Technical Report 18, Forest Science Monograph. 2

Leuschner, W. (1990). *Forest Regulation, Harvest Scheduling, and Planning Techniques* (First ed.). New York: Wiley. 281p. 5

Lockwood, C. and T. Moore (1993). Harvest scheduling with spatial constraints: a simulated annealing approach. *Canadian Journal of Forest Research 23*, 468–478. 2, 4

Metropolis, N., A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller (1953). Equation of state calculations by fast computing machines. *J. Chem. Physics 21*, 1087–1091. 2

Mullen, D. and R. Butler (1997, May). The design of a genetic algorithm based spatially constrained timber harvest scheduling model. In *Seventh Symposium on Systems Analysis in Forest Resources*, Number NC-205 in Gen. Tech. Rep., Bellaire, Michigan, pp. 1–5. U.S. Department of Agriculture, Forest Service, North Central Research Station. 2

Van Deusen, P. C. (1999). Multiple solution harvest scheduling. *Silva Fennica 33*(3), 207–216. 2, 3, 5

Van Deusen, P. C. (2001). Scheduling spatial arrangement and harvest simultaneously. *Silva Fennica 35*(1), 85–92. 2, 3, 5

5 Tables

Table 1: Example 1: Linear programming regime assignment summary.

Regime	N	x
16	37	1.00
1	37	0.96
2	22	0.92
3	15	0.91
4	16	0.97
5	18	0.92
6	15	0.91
7	19	0.90
8	19	0.95
9	25	0.92
10	26	0.94
11	26	0.93
12	32	0.91
13	34	0.93
14	42	0.98
15	59	0.98

Table 2: Example 2: Linear programming regime assignment summary.

Regime	N	x
CC1	2	1.00
CC2	7	0.87
CC3	10	0.92
CC4	10	0.96
CC5	9	0.82
CC6	10	0.93
CC7	14	0.95
CC8	16	0.92
CC9	24	0.95
CC10	16	0.96
CC11	25	0.94
CC12	21	0.98
CC13	36	0.96
CC14	51	0.98
CC15	43	1.00
T1,14	14	0.96
T2,14	5	0.92
T5,15	72	1.00
T7	2	1.00

Table 3: Categorized data for Example 3

Type	Count	Acres	Description
Bareland	80	2208	Current cutover polygons
Natural Hardwood	369	7620	
Natural Pine	29	98.5	Loblolly
Non-productive	21	123	Gullies, cemetaries, etc.
Open	37	32	Wildlife patches, non-forested yet productive
Planted Pine	382	11469	Loblolly
Pond	12	61	
Row	66	234	Roads, gas and power lines, railroads, etc.
Seeded Pine	19	145	Loblolly
Swamp	31	253	

Table 4: Example 3: Linear programming regime assignment summary.

Regime	N	x
CC2	19	1.00
CC3	53	0.97
CC4	15	0.91
CC5	21	0.95
CC6	31	0.98
CC7	25	0.94
CC8	22	0.96
CC9	25	0.97
CC10	29	0.98
CC11	10	0.85
CC12	12	0.97
CC13	16	0.89
CC14	27	0.98
CC15	34	0.97
CC16	31	0.96
CC17	28	0.99
CC18	10	0.91
CC19	8	0.98
CC20	1	1.00
T1,17	82	0.99
T2,17	40	0.99
T2,18	35	0.98
T3,18	27	1.00
T3,19	6	1.00
T4,19	1	1.00
T4,20	68	1.00
T5,20	17	1.00
T6,20	14	1.00
T7,20	21	1.00
T8,20	27	1.00
T9,20	23	1.00
T10,20	21	1.00
BARE1,18	19	1.00
BARE1,19	38	1.00
BARE1,20	24	0.97

6 Figures

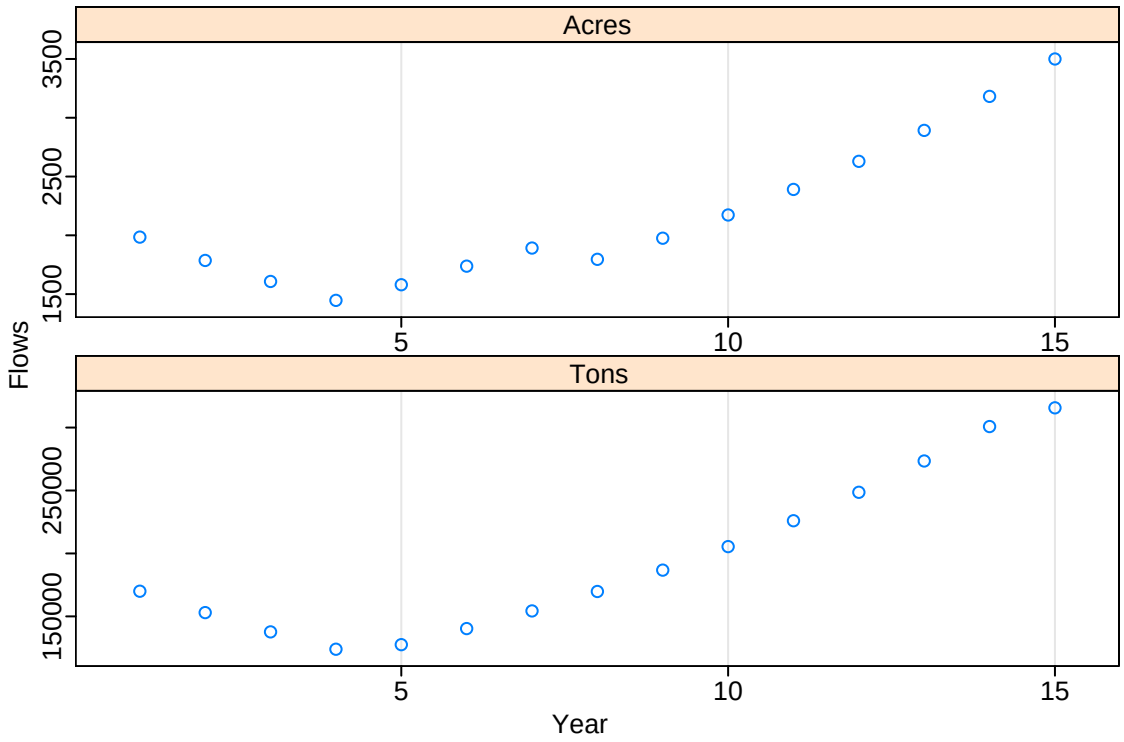

Figure 1: Example 1: Linear Programming annual cut in acres and tons. Objective function = 548,673

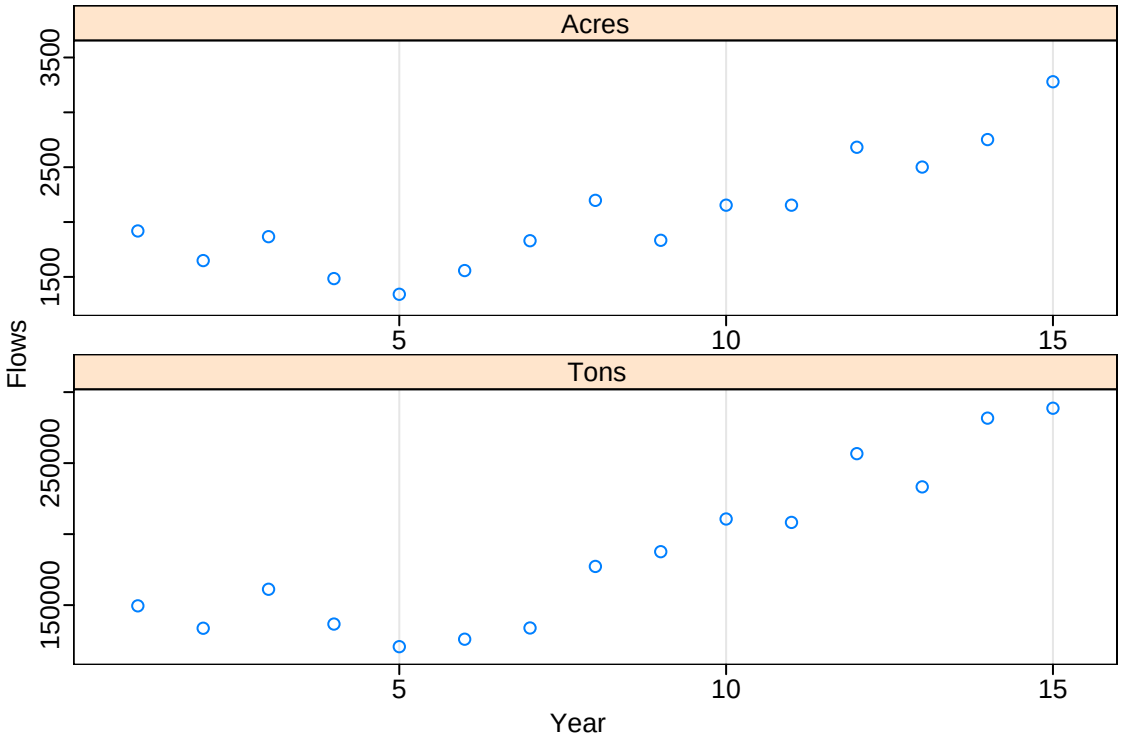

Figure 2: Example 1: Metropolis annual cut in acres and tons. Objective function = 535,164.

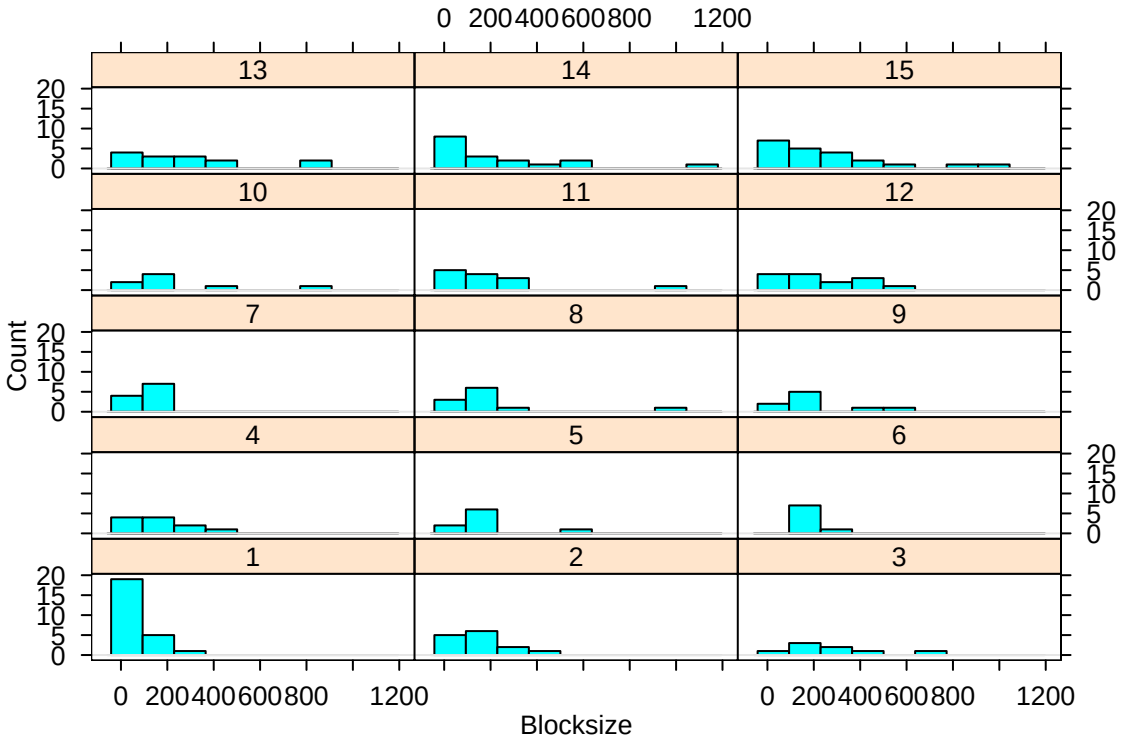

Figure 3: Example 1: Block size distribution from linear programming solution.

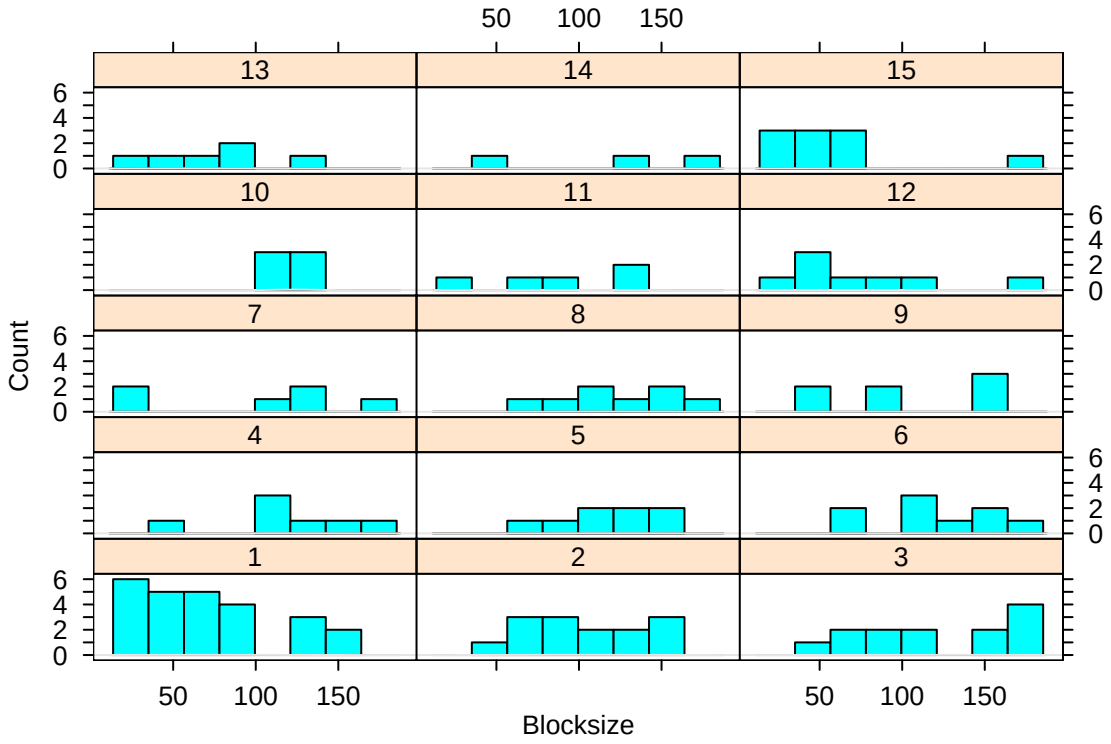

Figure 4: Example 1: Block size distribution from spatially constrained solution.

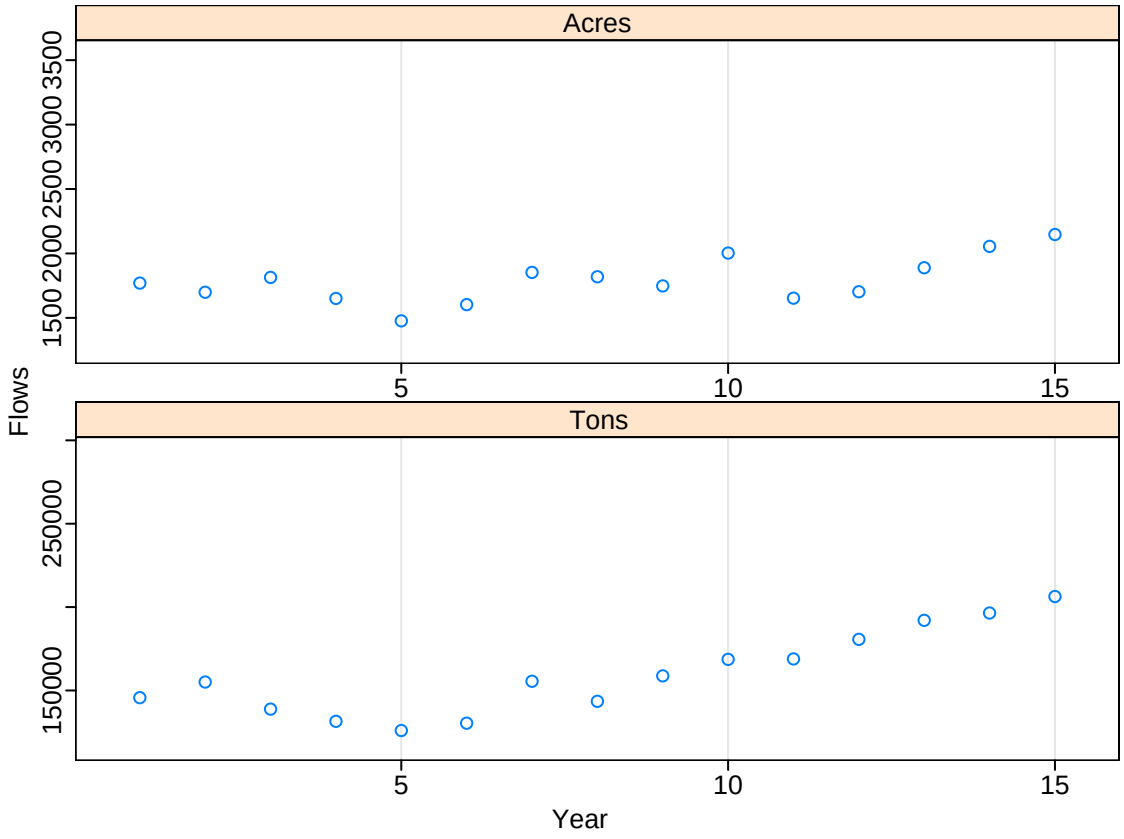

Figure 5: Example 1: Spatially constrained cut in acres and tons. Objective function = 442,306.

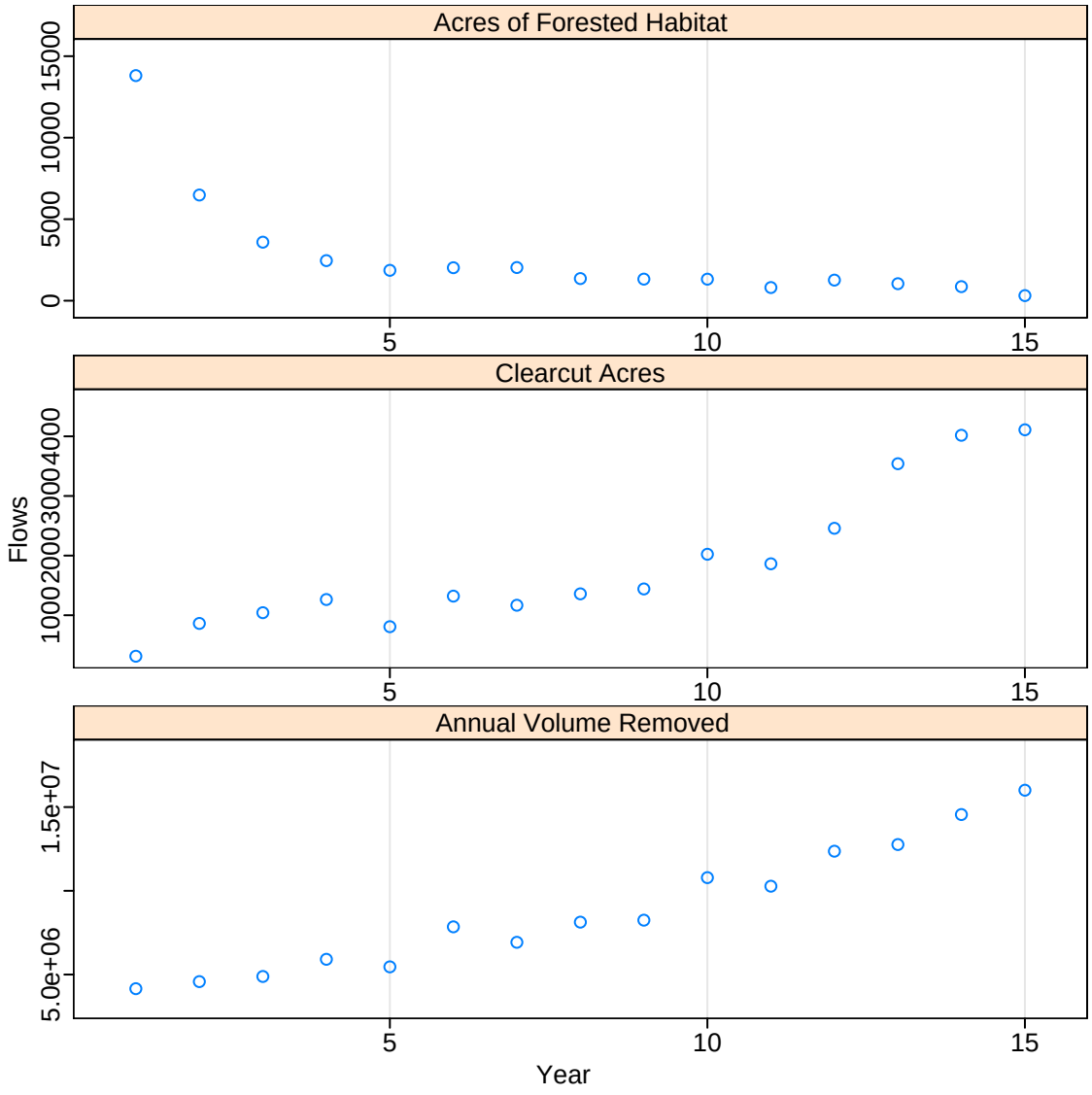

Figure 6: Example 2: Annual trends from linear programming solution. Objective function = 1.35e+08.

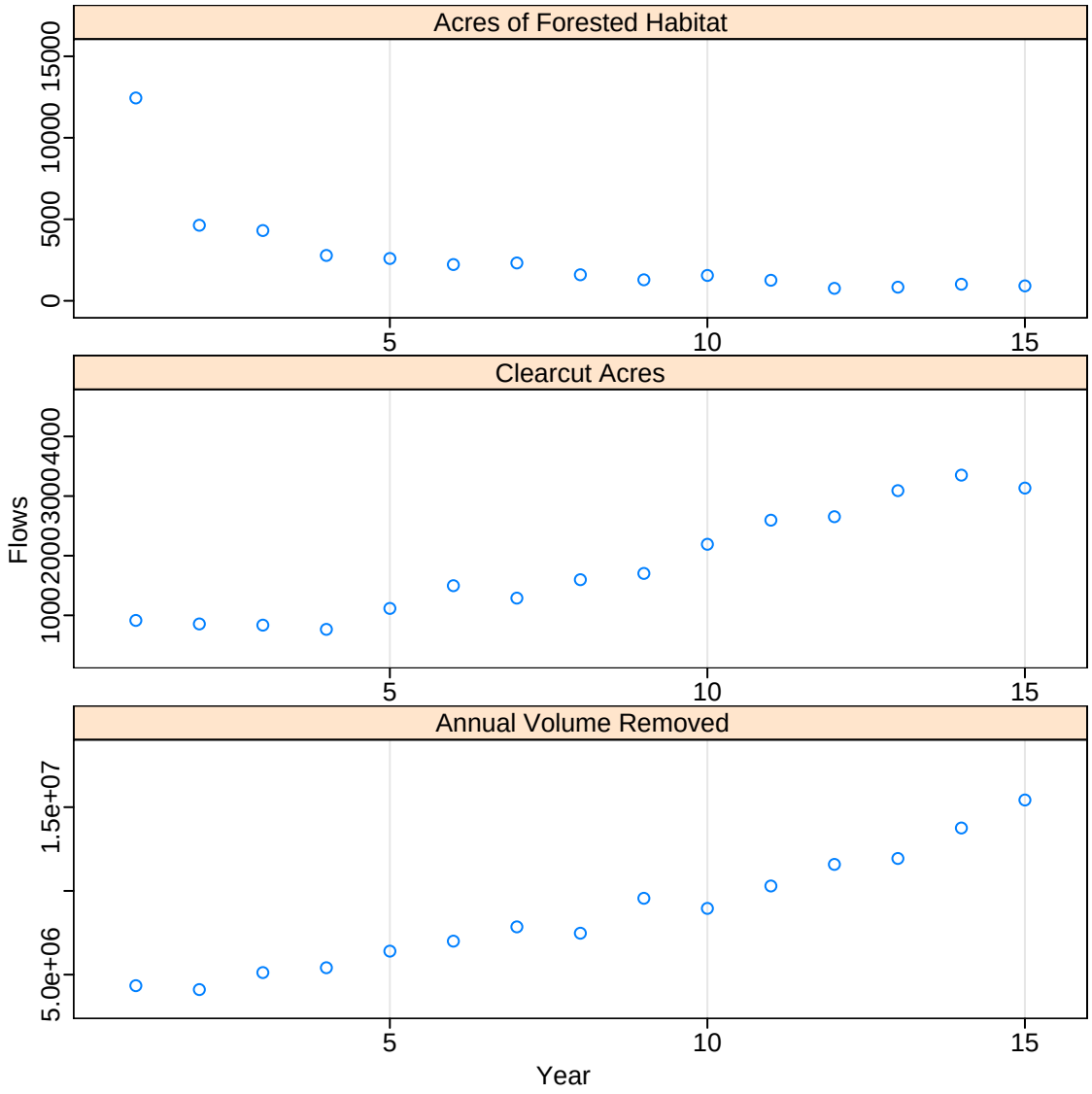

Figure 7: Example 2: Annual trends from spatially unconstrained Metropolis solution. Objective function = 1.32e+08.

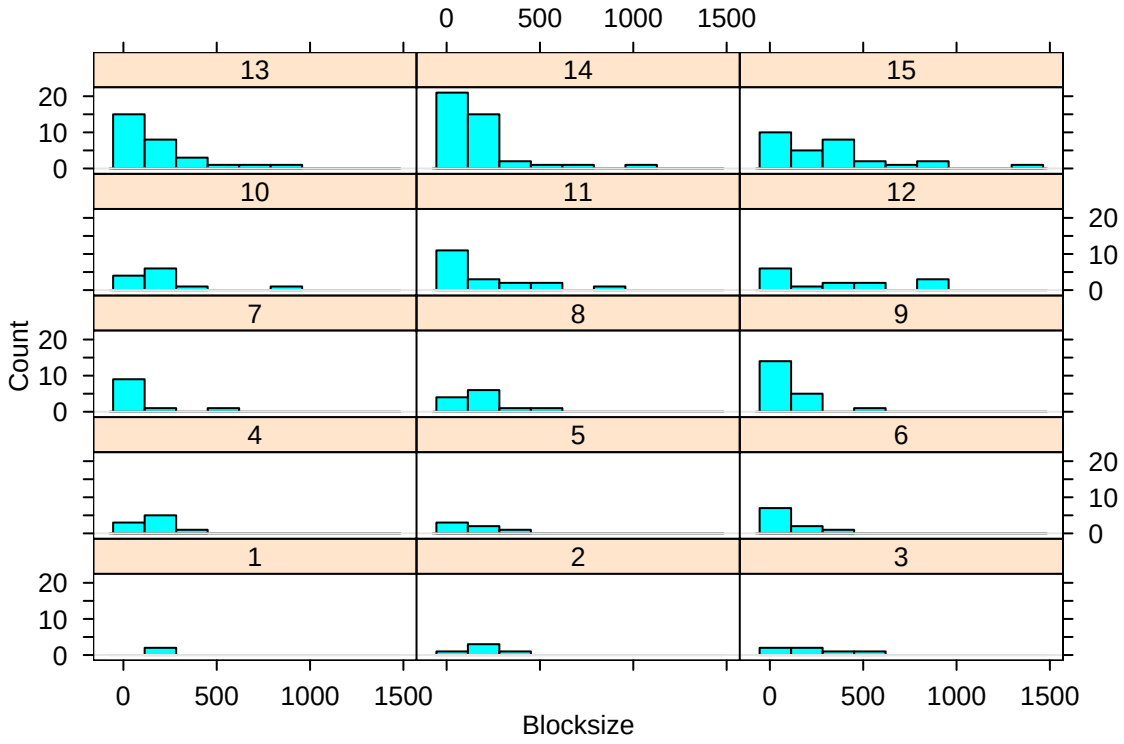

Figure 8: Example 2: Blocksize distribution from linear programming solution.

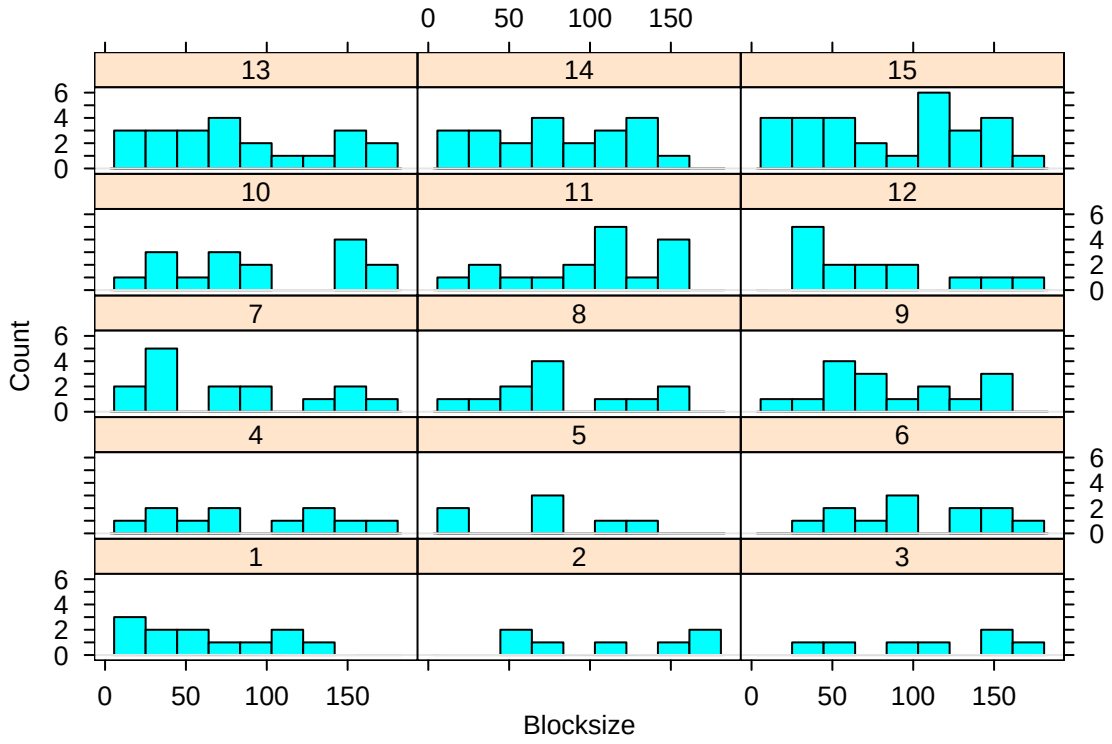

Figure 9: Example 2: Blocksize distribution from spatially constrained Metropolis solution.

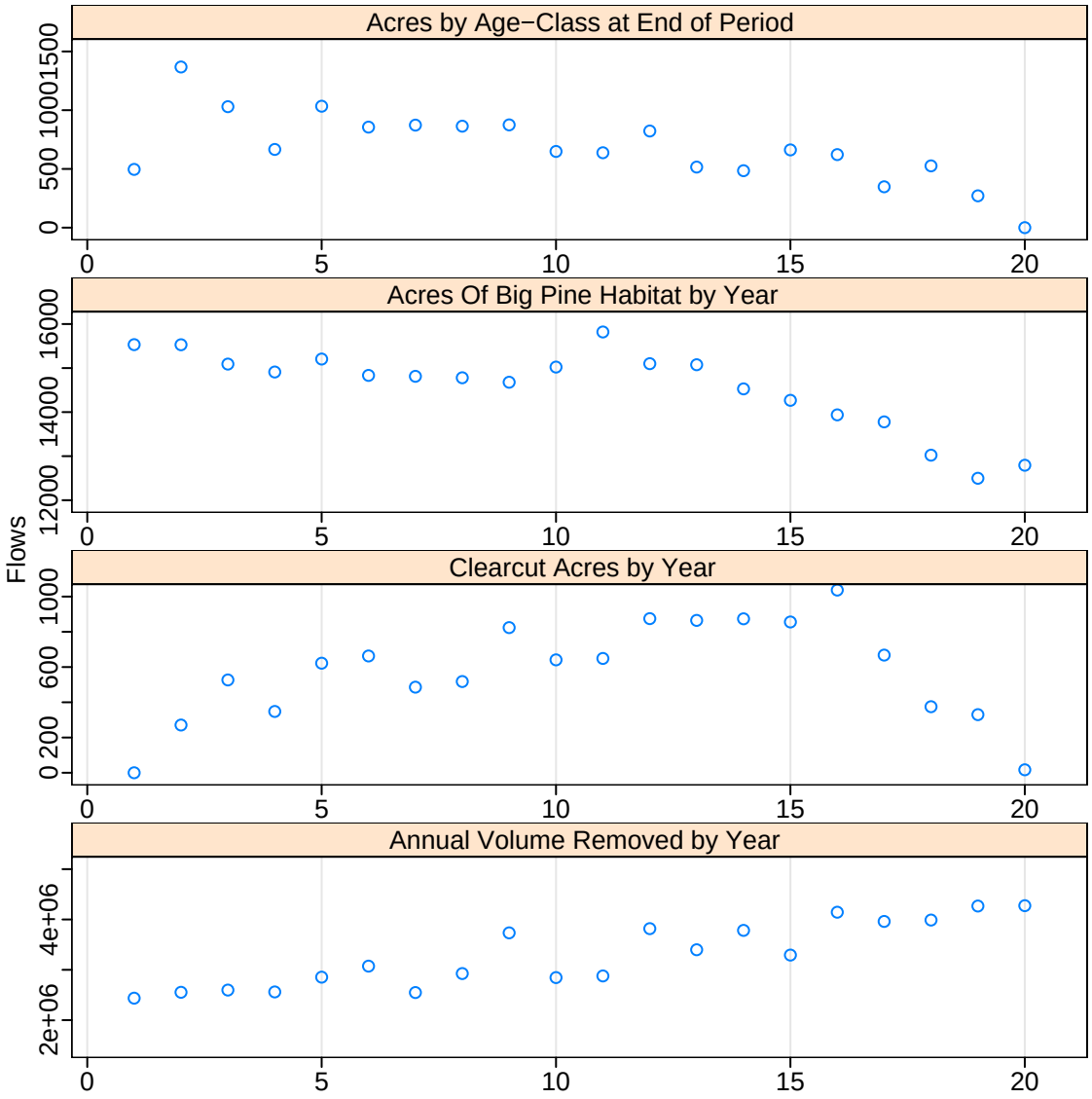

Figure 10: Example 3: Annual trends from linear programming solution. Objective function = 6.59e+07.

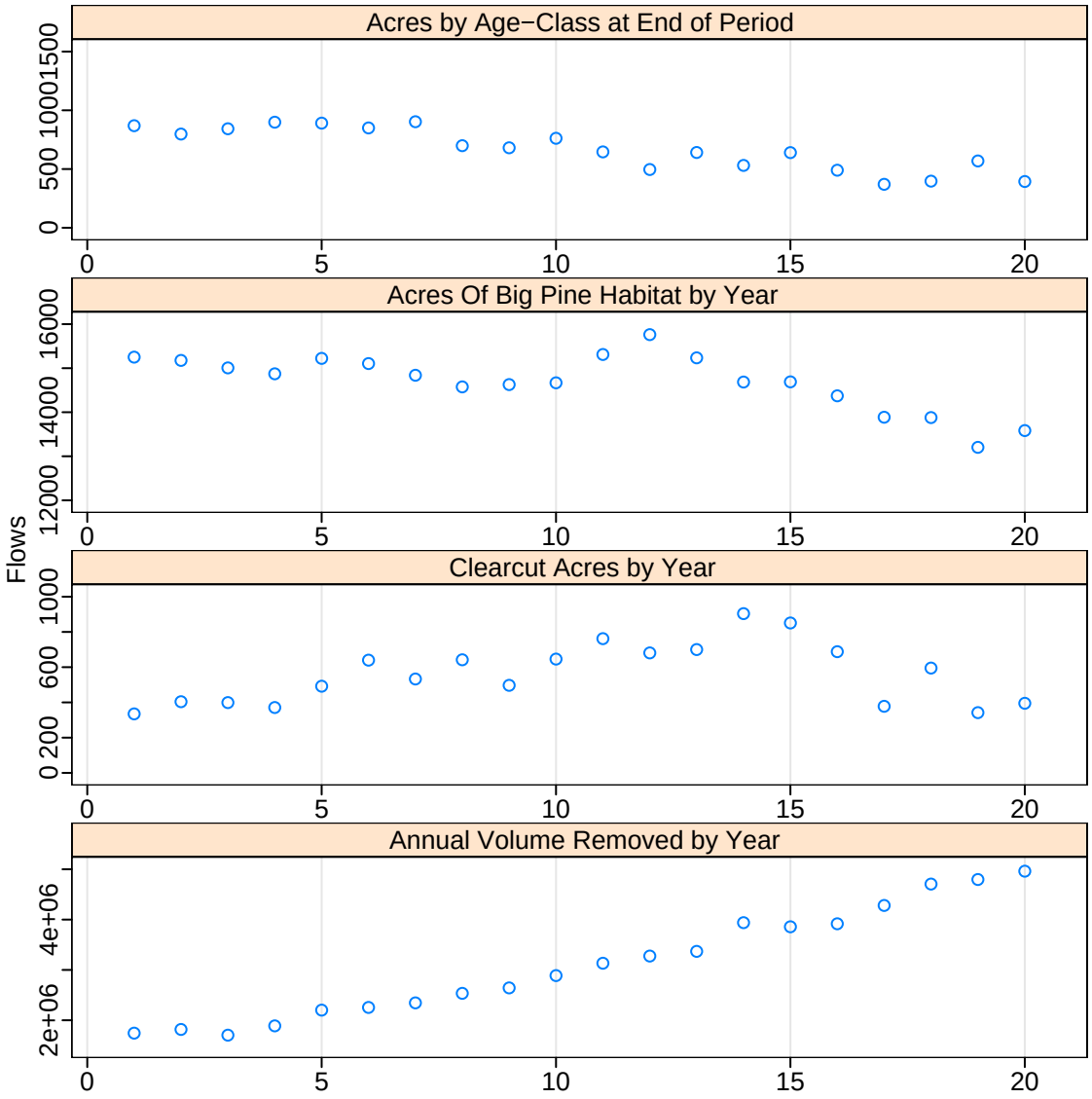

Figure 11: Example 3: Annual trends from spatially unconstrained Metropolis solution. Objective function = 6.15e+07.

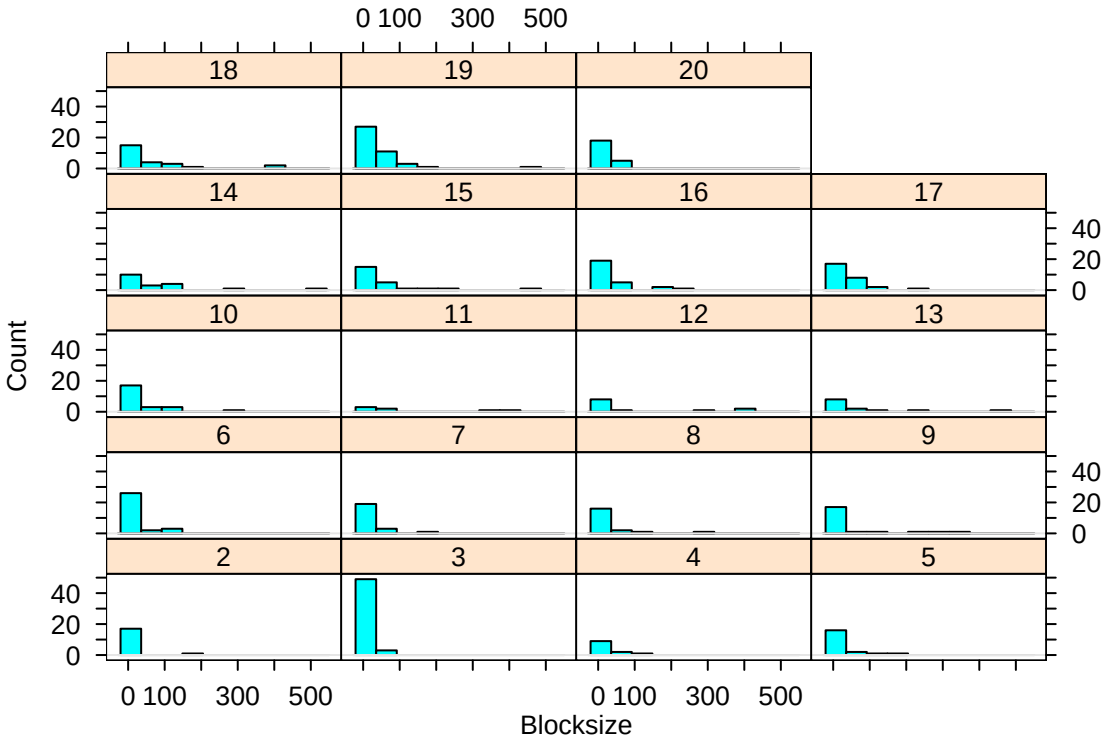

Figure 12: Example 3: Blocksize distribution from linear programming solution

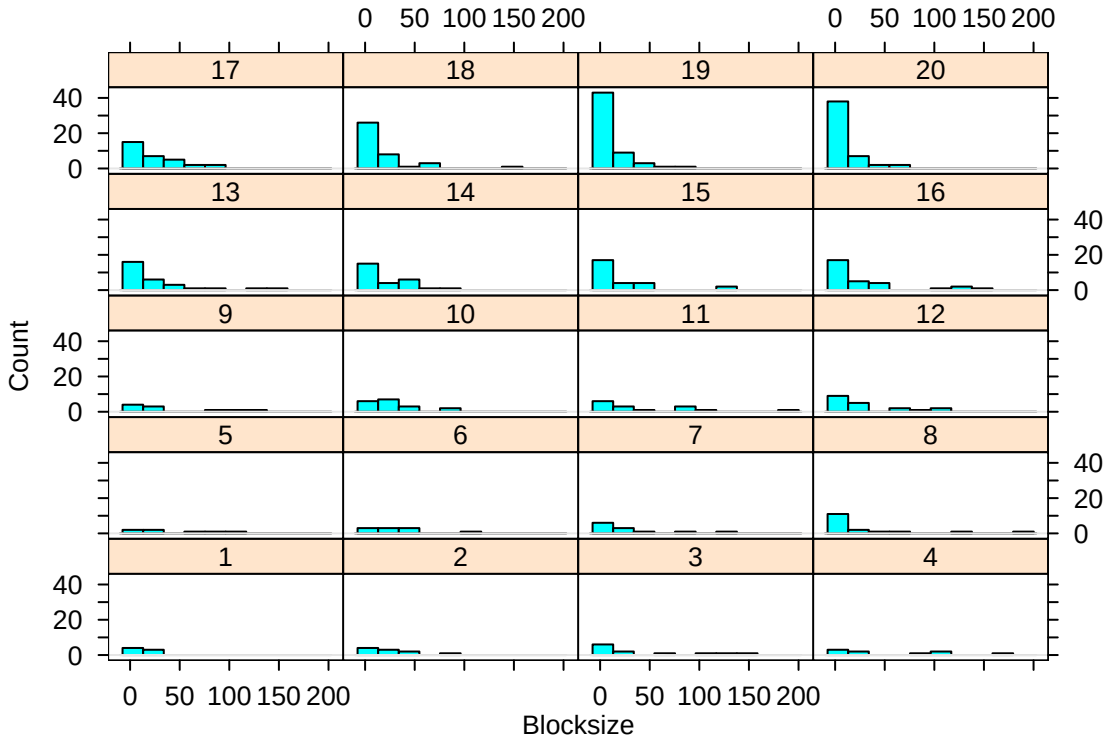

Figure 13: Example 3: Blocksize distribution from spatially constrained Metropolis solution.