Taylor & Francis
Taylor & Francis Group

Scandinavian Journal of Forest Research

ISSN: 0282-7581 (Print) 1651-1891 (Online) Journal homepage: http://www.tandfonline.com/loi/sfor20

A linear programming algorithm and software for forest-level planning problems including factories

Juha Lappi & Reetta Lempinen

To cite this article: Juha Lappi & Reetta Lempinen (2014) A linear programming algorithm and software for forest-level planning problems including factories, Scandinavian Journal of Forest Research, 29:sup1, 178-184, DOI: 10.1080/02827581.2014.886714

To link to this article: http://dx.doi.org/10.1080/02827581.2014.886714

Accepted online: 23 Jan 2014.Published online: 20 Feb 2014.

Submit your article to this journal

Article views: 100

View related articles

View Crossmark data

Citing articles: 1 View citing articles

Full Terms & Conditions of access and use can be found at
http://www.tandfonline.com/action/journalInformation?journalCode=sfor20

Download by: [Luke Natural Resources Institute Finland] **Date:** 01 October 2015, At: 04:06

Scandinavian Journal of Forest Research, 2014
Vol. 29, No. Supplement 1, 178–184, http://dx.doi.org/10.1080/02827581.2014.886714

Taylor & Francis
Taylor & Francis Group

RESEARCH ARTICLE

A linear programming algorithm and software for forest-level planning problems including factories

Juha Lappi[a*] and Reetta Lempinen[b]

[a]*Finnish Forest Research Institute, Suonenjoki Research Unit, Juntintie 154, Suonenjoki FI-77600, Finland;* [b]*Finnish Forest Research Institute, Joensuu Research Unit, Box 68, Joensuu FI 80101, Finland*

(*Received 28 March 2013; accepted 19 January 2014*)

Combining stand simulation and forest-level optimization is an efficient way to study harvest scenarios of a forest area. A simulator first generates for each treatment unit a number of treatment schedules. Linear programming (LP) can then be used to study how stand-level schedules can be combined at the forest level with respect to alternative goals and constraints. The special structure of the obtained LP problems can be utilized using the generalized upper-bound technique which takes care of the so-called area constraints. JLP software was based on this technique. Later J software was developed to replace JLP. Now J is developed to deal with factory problems where the transportations costs and capacities of factories are included in the problem definition. The generalized upper-bound technique was modified to handle transportation constraints which tell that each timber unit produced is transported to some of the factories. The number of these constraints is very large. This paper describes the basic features of the algorithm and its implementation in the J software.

Keywords: forest planning; linear programming; software development; transportation cost

Downloaded by [Luke Natural Resources Institute Finland] at 04:06 01 October 2015

Introduction

Combining stand simulation and forest-level optimization is a robust and efficient way to study harvest scenarios of a forest area. A simulator first generates for each treatment unit (stand) a number of treatment schedules which make sense from the viewpoint of stand management. Linear programming (LP) can then be used to study how stand-level schedules can be combined at the forest level in an efficient way with respect to alternative goals and constraints. This approach has been used, e.g. in FORPLAN (Turner & Church 1995), MELA (Redsven et al. 2012), GAYA-JLP (Hoen 1996), SPECTRUM (Camenson 1996), SIMO (Kangas & Rasinmäki 2008; Rasinmäki et al. 2009), and DTRAN (Hoganson & Kapple 1991).

Model I type of the LP formulation (see, e.g. Dykstra 1984) keeps the identity of management units over time, while in Model II, units regenerated at the same time are combined. FORPLAN and SPECTRUM use Model II formulation, and other mentioned systems apply Model I formulation. The special structure of the Model I LP problems can be utilized using the generalized upper-bound (GUB) technique of Dantzig and VanSlyke (1967) which can take care of the so-called area constraints without having them as a part of the basis matrix. JLP software (Lappi 1992) was designed to solve forest management problems using this technique. JLP is used in the Finnish MELA system and was used in the GAYA system. Later J software was developed to replace JLP. Software was revised with new linear algebra subroutines and dynamic memory allocation. In addition to LP, J contains additional features such as statistical functions, stem curves and simulator language. J is currently used in the Norwegian GAYA system (Bergseng et al. 2012) and in the Finnish SIMO system (Kangas & Rasinmäki 2008).

Neither JLP nor original J could handle planning problems including factories. In a factory problem, the transportations costs and capacities of factories are included in the problem definition. For instance, the net present value can be maximized subject to capacity constraints and sustainability constraints. Factory problems considered here are more simple than many factory location problems analyzed in supply chain models where the production process of factories are analyzed in a greater detail (see e.g. Vila et al. 2006; Melo et al. 2009). The main emphasis is to analyze strategic timber management options while taking into account factory locations and capacities (see Gunn 2009).

DTRAN is solving approximately LP problems utilizing the dual problem (Hoganson & Rose 1984). The computations of DTRAN are very similar to the computations done in JLP/J. In JLP and J, the next

*Corresponding author. Email: juha.lappi@metla.fi

© 2014 Taylor & Francis

Downloaded by [Luke Natural Resources Institute Finland] at 04:06 01 October 2015

optimal schedule in a treatment unit is selected in a similar way as in DTRAN, using the shadow prices of forest variables. The difference is that in DTRAN, these prices are heuristically obtained approximate prices, while in JLP and J, the prices are computed exactly according to the LP theory. DTRAN can handle problems with factories.

Now the J software is developed to be able to solve factory problems (Lappi & Lempinen 2013). First, the GUB technique was modified to handle transportation constraints which tell that each timber assortment unit produced in a given unit is transported to some of the factories. The number of these constraints is proportional to a number of timber assortments × number of periods × number of units, that is, generally very large. Then this algorithm was implemented in the J software. This paper described the basic features of the mathematical derivation of the algorithm. Some details of the implementation are given. An example is presented.

Optimization problem

Mathematically, the optimization problems considered can be defined as follows:

$$\text{Max or Min}\, z_0 = \sum_{k=1}^{p} a_{0k}x_k + \sum_{k=1}^{q} b_{0k}z_k + \sum_{k=1}^{p}\sum_{f=1}^{F} \alpha_{0kf}x_{kf} + \sum_{k=1}^{p}\sum_{f=1}^{F} \beta_{0kf}y_{kf} \quad (1)$$

subject to the following utility constraints (note that objective row is formulated as utility constraint $t = 0$):

$$c_t \leq \sum_{k=1}^{p} a_{tk}x_k + \sum_{k=1}^{q} b_{tk}z_k + \sum_{k=1}^{p}\sum_{f=1}^{F} \alpha_{tkf}x_{kf} + \sum_{k=1}^{p}\sum_{f=1}^{F} \beta_{tkf}y_{kf} \leq C_t, \quad t = 1,\ldots,r \quad (2)$$

Technical constraints are the following:

$$x_k - \sum_{i=1}^{m}\sum_{j=1}^{n_i} x_k^{ij} w_{ij} = 0, \quad k = 1,\ldots,p \quad (3)$$

$$\sum_{j=1}^{n_i} w_{ij} = A_i, \quad i = 1,\ldots,m \quad (4)$$

$$x_{kf} - \sum_{i=1}^{m} x_{kf}^{i} = 0, \quad (k,f) \in \mathbf{R} \quad (5)$$

$$y_{kf} - \sum_{i=1}^{m} \gamma_{kf}^{i} x_{kf}^{i} = 0, \quad (k,f) \in \mathbf{B} \quad (6)$$

$$\sum_{f=1}^{F} x_{kf}^{i} - \sum_{j}^{n_i} x_k^{ij} w_{ij} = 0, \quad i = 1,\ldots,m, \quad k \in \mathbf{K} \quad (7)$$

$$w_{ij} \geq 0, i = 1,\ldots,m, j = 1,\ldots n_i, z_k \geq 0 \quad \text{for}\, k = 1,\ldots,q,$$
$$x_{kf}^{i} \geq 0, (k,f) \in \mathbf{R}, x_k^{ij} \geq 0, k \in \mathbf{K}$$

where m = number of treatment units; n_i = number of management schedules for unit i; w_{ij} = the area of the treatment unit i managed according to management schedule j; x_k^{ij} = amount per unit area of item (commodity) k produced by unit i if schedule j is applied (constants produced by the treatment simulator); x_k = obtained amount of item k, k = 1, …, p, z_k= an additional decision variable, k = 1, …, q (e.g. slack and surplus variables in goal programming); a_{tk} = fixed real constants for t = 0, …, r, k = 1, …, p; b_{tk} = fixed real constants for t = 0, …, r, k = 1, …, q; α_{tkf}= fixed real constants for t = 0, …, r, k = 1, …, p, f = 1, …, F; β_{tkf}= fixed real constants for t = 0, …, r, k = 1, …, p, f = 1, …, F; r = number of utility constraints, x_{kf}^{i}= amount of item k transported from unit i to factory f; y_{kf} = utility obtained when item k is transported to factory f (the transportation cost is taken into account); γ_{kf}^{i}= utility when one unit of item k is transported from unit i to factory f (the transportation cost is taken into account); F = number of factories; A_i = area of unit i; $\mathbf{R}$ = set of (k, f) such that $\alpha_{tkf} > 0$ or $\beta_{tkf} > 0$ for some t; $\mathbf{B}$ = set of (k, f) such that $\beta_{tkf} > 0$ for some t; $\mathbf{K}$ = set of such k that $\alpha_{tkf} > 0$ or $\beta_{tkf} > 0$ for some t and f.

Note that time does not appear in the problem definition. Time is taken into account implicitly in the item k; e.g. item k refers to harvested pulp wood at second subperiod. Equation (3) defines the aggregated variable x_k as the sum over all units and schedules. Equation (4) (area constraint) states that the areas under different schedules add up to the total area of the unit. If constants x_k^{ij} are expressed as the total amount in the unit (instead per area), then w_{ij}'s are proportions and each area A_i is 1. Equation (5) states that item k assigned to factory f is obtained by adding up all unit-wise assignments. Equation (6) tells that utility of item k when transported to factory f is obtained by summing up unit-wise utilities. Equation (7) (transportation constraint) tells that all of item k in unit i are transported to factories. Note that constraint $x_k^{ij} \geq 0$, $k \in \mathbf{K}$ (stating that transported items are nonnegative) is not standard LP constraint because it constrains the values of the problem coefficients, not variables. Note that taking into account Equation (6), $\beta_{tkf}y_{kf} = \sum_{i=1}^{m} \beta_{tkf}\gamma_{kf}^{i}x_{kf}^{i}$. Thus multiplying γ_{kf}^{i} for each k and f by a constant and dividing each β_{tkf} by the same constant, we can get an equivalent problem. Thus we can assume without loss of generality that each β_{tkf} is one. This assumption is made in J, but the formulas are presented below without this assumption. A typical value of a_{tk} or b_{tk} is 1 or –1 for the constraint rows ($t > 0$). If the net present value is maximized, α_{0kf} is the discounted factory price of item k in factory f.

Usually y_{kf}-variables appear in the objective row as a part of the definition of the net present value. When trying to understand the formulas, it might be easier to consider that y_{kf} is the total discounted transportation cost for item k and γ^i_{kf} is the discounted per unit transportation cost when item k is transported from unit i to factory f. Also when the utility of having item k transported to factory f (discounted factory price) is taken into account in y_{kf} and γ^i_{kf}, we get more efficient computations and more compact problem definition in J. Note that we can have different factory groups for different timber assortments by setting properly zeroes to alfas and betas.

In typical problems, the utility constraints including x_{kf} are of form $x_{kf} \le C$, which states that the capacity of factory f has an upper bound C for a period-specific timber assortment. There can also be a lower bound stating a minimum demand.

It is assumed that the decision-maker is interested in the objective function and in the utility constraints which together define the utility model of the decision-maker. That is, it is assumed that the decision-maker is interested only in total amounts, not in the way total amounts are obtained by summing over stands or transportation routes. The user of J must only specify the objective (1) and the utility constraints (2) and give information how the program can compute or access coefficients γ^i_{kf}. The program takes care automatically of the technical constraints (3)–(7).

Solution algorithm

The above formulation is logical for describing the problem. The solution algorithm is based on rewriting the problem. The area constraints (4) and transportation constraints (7) are taken care using the GUB technique, and the definitions of x_k-variables and factory variables x_{kf} and y_{kf} are written directly into the objective row (1) and constraints (2). Let us assume that the objective is to be maximized (minimization can be obtained by maximizing the negative of the objective function).

Constraint (4) will be automatically satisfied if we select from each unit i a schedule $J(i)$, called the key schedule, and write $w_{iJ(i)}$ in terms of the other weights:

$$w_{iJ(i)} = A_i - \sum_{j \neq J(i)} w_{ij} \tag{8}$$

Constraint (3) can then be written as

$$\begin{aligned} x_k &= \sum_{i=1}^{m}\sum_{j=1}^{n_i} x_k^{ij} w_{ij} = \sum_{i=1}^{m}\sum_{j\neq J(i)}^{n_i} x_k^{ij} w_{ij} + \sum_{i=1}^{m} x_k^{iJ(i)} w_{iJ(i)} \\ &= \sum_{i=1}^{m}\sum_{j\neq J(i)} \left(x_k^{ij} - x_k^{iJ(i)}\right) w_{ij} + \sum_{i=1}^{m} A_i x_k^{iJ(i)} \end{aligned} \tag{9}$$

Computations utilize directly this definition in Equations (1) and (2). The algorithm takes care that at each stage $w_{iJ(i)} > 0$. Similarly, we can apply the GUB technique in Equation (7) by selecting for each unit i and transported item k (i.e $k \in \mathbf{K}$) a key factory $f^*(i,k)$. Using Equation (7) and after separating $w_{iJ(i)}$, we can write

$$\begin{aligned} x^i_{kf^*(i,k)} &= -\sum_{f\neq f^*(i,k)} x^i_{kf} + \sum_{j=1}^{n_i} x_k^{ij} w_{ij} \\ &= -\sum_{f\neq f^*(i,k)} x^i_{kf} + \sum_{j\neq J(i)} \left(x_k^{ij} - x_k^{iJ(i)}\right) w_{ij} + A_i x_k^{iJ(i)} \end{aligned} \tag{10}$$

At each stage $x^i_{f^*(i,k)k} > 0$. Constraint (6) can then be written by separating $x^i_{f^*(i,k)k}$ and using Equation (10):

$$\begin{aligned} y_{kf} &= \sum_{i=1}^{m} \gamma^i_{kf} x^i_{kf} = \sum_{i:f\neq f^*(i,k)} \gamma^i_{kf} x^i_{kf} + \sum_{i:f=f^*(i,k)} \gamma^i_{kf^*(i,k)} x^i_{kf^*(i,k)} \\ &= \sum_{i:f\neq f^*(i,k)} \gamma^i_{kf} x^i_{kf} + \sum_{i:f=f^*(i,k)} \gamma^i_{kf^*(i,k)} \\ &\quad \left(-\sum_{f'\neq f^*(i,k)} x^i_{kf'} + \sum_{j\neq J(i)} \left(x_k^{ij} - x_k^{iJ(i)}\right) w_{ij} + A_i x_k^{iJ(i)}\right) \\ &= \sum_{i:f\neq f^*(i,k)} \gamma^i_{kf} x^i_{kf} - \sum_{i:f=f^*(i,k)} \sum_{f'\neq f^*(i,k)} \gamma^i_{kf^*(i,k)} x^i_{kf'} \\ &\quad + \sum_{i:f=f^*(i,k)} \sum_{j\neq J(i)} \gamma^i_{kf^*(i,k)} \left(x_k^{ij} - x_k^{iJ(i)}\right) w_{ij} \\ &\quad + \sum_{i:f=f^*(i,k)} \gamma^i_{kf^*(i,k)} A_i x_k^{iJ(i)} \end{aligned} \tag{11}$$

x_{kf} is a special case of y_{kf} obtained by letting $\gamma^i_{kf} = 1$.

We can simplify the formulas and the computations by using "one-x formulation" of Lappi (1992), i.e. by defining $x_t^{ij} = \sum_{k=1}^{p} a_{tk} x_k^{ij}$ (this defines a new indexing of x's). x_t^{ij}'s are computed in advance as temporary forest variables. The objective row or the constraint row of the modified problem will then be as follows:

Downloaded by [Luke Natural Resources Institute Finland] at 04:06 01 October 2015

Downloaded by [Luke Natural Resources Institute Finland] at 04:06 01 October 2015

$$
\begin{aligned}
L_t = {} & \sum_{k=1}^{p} a_{tk} \sum_{i=1}^{m} \Big[\sum_{j \neq J(i)} x_k^{ij} w_{ij} + x_k^{iJ(i)} (A_i - \sum_{j \neq J(i)} w_{ij}) \Big] + \sum_{k=1}^{q} b_{tk} z_k \\
& + \sum_{k=1}^{p} \sum_{f=1}^{F} \alpha_{tkf} \Big(\sum_{i:f \neq f^*(i,k)} x_{kf}^{i} - \sum_{i:f = f^*(i,k)} \sum_{f' \neq f^*(i,k)} x_{kf'}^{i} \\
& + \sum_{i:f = f^*(i,k)} \sum_{j \neq J(i)} \left(x_k^{ij} - x_k^{iJ(i)} \right) w_{ij} + \sum_{i:f = f^*(i,k)} A_i x_k^{iJ(i)} \Big) \\
& + \sum_{k=1}^{p} \sum_{f=1}^{F} \beta_{tkf} \Big(\sum_{i:f \neq f^*(i,k)} \gamma_{kf}^{i} x_{kf}^{i} - \sum_{i:f = f^*(i,k)} \sum_{f' \neq f^*(i,k)} \gamma_{kf^*(i,k)}^{i} x_{kf'}^{i} \\
& + \sum_{i:f = f^*(i,k)} \sum_{j \neq J(i)} \gamma_{kf^*(i,k)}^{i} (x_k^{ij} - x_k^{iJ(i)}) w_{ij} \\
& + \sum_{i:f = f^*(i,k)} \gamma_{kf^*(i,k)}^{i} A_i x_k^{iJ(i)} \Big) = \sum_{k=1}^{q} b_{tk} z_k \\
& + \sum_{i=1}^{m} \sum_{j \neq J(i)} \Big((x_t^{ij} - x_t^{iJ(i)}) + \sum_{k=1}^{p} \Big(\alpha_{tkf^*} \\
& + \beta_{tkf^*(i,k)} \gamma_{kf^*(i,k)}^{i} \Big) (x_k^{ij} - x_k^{iJ(i)}) \Big) w_{ij} \\
& + \sum_{i=1}^{m} \sum_{k=1}^{p} \sum_{f \neq f^*(i,k)} \Big(\alpha_{tkf} - \alpha_{tkf^*(i,k)} \\
& + \beta_{tkf} \gamma_{kf}^{i} - \beta_{tkf^*(i,k)} \gamma_{kf^*(i,k)}^{i} \Big) x_{kf}^{i} + \sum_{i=1}^{m} A_i x_t^{iJ(i)} \\
& + \sum_{i=1}^{m} \sum_{k=1}^{p} \Big(\alpha_{tkf^*(i,k)} + \beta_{tkf^*(i,k)} \gamma_{kf^*(i,k)}^{i} \Big) A_i x_k^{iJ(i)} \qquad (12)
\end{aligned}
$$

The derivation of this equation is the most difficult part of the algorithm. For given key schedules and key factories, the unknown variables in Equation (12) are $w_{ij}, j \neq J(i)$, $x_{kf}^{i},\ f \neq f^*(i,k)$, and z_k, k = 1, …, q. The columns of the working basis matrix are formed from their coefficients for basic variables. If $w_{ij}, j \neq J(i)$ is a basic variable, then the corresponding schedule j is called an explicit basic schedule. Lower and upper bounds are treated simultaneously so that at each point either the lower or the upper bound is active. The dimension of the working basis matrix is equal to r, i.e. the number of utility constraints. If definitions of $x_k, x_{kf},$ and y_{kf} (Equations (3), (5) and (6)) are written into Equations (1), (2), and (7), and if separate constraints are formed for c_t and C_t, the dimension of the basis matrix in a standard LP algorithm is $2r + m + m|\mathbf{K}|$, i.e. very large.

For given key schedules and key factories, the last two terms in Equation (12) are constants, and they need to be transferred to the RHS in the constraint rows.

Revised simplex method

The theory of revised simplex method needed here can be presented as follows: (see, e.g. Luenberger & Ye 2008). Let the LP problem be the following:

$$\max \mathbf{c}'\mathbf{y} \qquad (13)$$

$$\text{s.t. } \mathbf{A}\mathbf{y} = \mathbf{b} \qquad (14)$$

Let the problem be decomposed as

$$\max\ \mathbf{c}_b'\mathbf{y}_b + \mathbf{c}_d'\mathbf{y}_d \qquad (15)$$

$$\text{s.t. } \mathbf{B}\mathbf{y}_b + \mathbf{D}\mathbf{y}_d = \mathbf{b} \qquad (16)$$

where $\mathbf{B}$ is the current basis. Giving new values for nonbasic variables $\mathbf{y}_d$ will result in new values of current basic variables

$$\mathbf{y}_b^{+} = \mathbf{B}^{-1}\mathbf{b} - \mathbf{B}^{-1}\mathbf{D}\mathbf{y}_d = \mathbf{y}_b - \mathbf{B}^{-1}\mathbf{D}\mathbf{y}_d \qquad (17)$$

The new objective value will be the following:

$$
\begin{aligned}
& \mathbf{c}_b'\mathbf{B}^{-1}\mathbf{b} + \left(\mathbf{c}_d' - \mathbf{c}_b'\mathbf{B}^{-1}\mathbf{D} \right)\mathbf{y}_d \\
& \quad = \mathbf{c}_b'\mathbf{y}_b + \left(\mathbf{c}_d' - \mathbf{c}_b'\mathbf{B}^{-1}\mathbf{D} \right)\mathbf{y}_d \qquad (18)
\end{aligned}
$$

Thus the cost vector $\left(\mathbf{c}_d' - \mathbf{c}_b'\mathbf{B}^{-1}\mathbf{D} \right)$ can be used to determine if new variables should enter into the basis. Write

$$\mathbf{v} = (v_1, \ldots, v_t, \ldots, v_r)' = \mathbf{c}_b'\mathbf{B}^{-1} \qquad (19)$$

When considering the update step where one column is added into the basis, $\mathbf{y}_d$ has one nonzero element, let it be denoted y_d and let $\mathbf{d}$ be the corresponding column of $\mathbf{D}$.

Leaving variable will be the first one becoming zero when increasing y_d. Let

$$\Delta = \mathbf{B}^{-1}\mathbf{d} \qquad (20)$$

The new value of an element k of $\mathbf{y}_b$ will become zero if Δ_k is positive and $y_d = y_{bk}/\Delta_k$, where y_{bk} is the current element. To obtain equality constraints, a residual variable is added into each constraint. The residual variables can be positive or negative.

Entering variable

Entering variable can be determined as follows. A new w_{ij} can enter the basis if

$$
\begin{aligned}
& \left(x_0^{ij} - x_0^{iJ(i)} \right) + \sum_{k=1}^{p} \Big(\alpha_{0kf^*(i,k)} \left(x_k^{ij} - x_k^{iJ(i)} \right) \\
& + \beta_{0kf^*(i,k)} \gamma_{kf^*(i,k)}^{i} \left(x_k^{ij} - x_k^{iJ(i)} \right) \Big) - \sum_{t=1}^{r} v_t \Big(\left(x_t^{ij} - x_t^{iJ(i)} \right) \\
& + \sum_{k=1}^{p} \Big(\alpha_{tkf^*(i,k)} \left(x_k^{ij} - x_k^{iJ(i)} \right) + \beta_{tkf^*(i,k)} \gamma_{kf^*(i,k)}^{i} \left(x_k^{ij} - x_k^{iJ(i)} \right) \Big) \Big) \\
& > 0 \qquad (21)
\end{aligned}
$$

Equation (21) is applied by computing the price of each schedule in a treatment unit. If the maximum value is greater that the value of the key schedule, then the weight of this optimal schedule will enter the working basis. If w_{ij} enters the basis, then the entering column can be seen from Equation (12). A z_k variable can enter the solution if $b_{0t} - \sum v_t b_{tk} > 0$. Element t of the entering column will b_{tk}

A x^i_{kf} variable can enter the basis if:

$$\alpha_{0kf} - \alpha_{0kf^*(i,k)} + \beta_{0kf}\gamma^i_{kf} - \beta_{0kf^*(i,k)}\gamma^i_{kf^*(i,k)} - \sum_{t=1}^{r} v_t\left(\alpha_{tkf} - \alpha_{tkf^*(i,k)} + \beta_{tkf}\gamma^i_{kf} - \beta_{tkf^*(i,k)}\gamma^i_{kf^*(i,k)}\right) > 0 \tag{22}$$

The element t of the corresponding entering column can be seen from Equation (12).

If the upper limit is binding, the negative residual variable can enter the solution if $v_t < 0$. Similarly, if the lower limit is binding, the positive residual variable can enter the solution, if $v_t > 0$.

Leaving variable

For all current basic variables, we first determine smallest $\lambda = \text{current value}/\Delta_k$ to see which variable might leave. We need take into account that it may become necessary to change key schedule or key factory, i.e. to consider that $w_{iJ(i)}$ or $x^i_{kf^*(i,k)}$ will become zero.

Key schedule needs to be changed if an implicit constraint $\sum_{j \neq J(i)} w_{ij} \leq A_i$ becomes binding for some i. Let function δ denote the elements of Δ corresponding to basic variables (e.g. $\delta(w_{ij})$ is the element of Δ corresponding to w_{ij}). If the entering variable is not w_{ij}, the implicit constraint will become binding if

$$\sum_{j \neq J(i)} w_{ij} - \lambda \sum_{j \neq J(i)} \delta(w_{ij}) = A_i \tag{23}$$

If the entering variable is $w_{ij'}$ we need to take into account the direct effect:

$$\sum_{j \neq J(i)} w_{ij} - \lambda \sum_{j \neq J(i)} \delta(w_{ij}) + \lambda = A_i \tag{24}$$

Equations (23) and (24) are solved for λ. If there are no explicit basic schedules in the unit i, then Equation (24) states that λ, i.e. the entering $w_{ij'}$ will get value A_i, i.e. we just need to change the current key schedule. Otherwise, we need to write the basis first in terms of some other key schedule. We need to update the RHS and write the new working basis in terms of the new key schedule.

We need to change the current key factory for variable k in unit i if $x^i_{kf^*(i,k)}$ defined with Equation (10) will become zero. If the entering variable is not x^i_{kf} and not $w_{ij'}$ then $x^i_{kf^*(i,k)}$ will be zero if

$$-\sum_{f \neq f^*(i,k)} x^i_{kf} + \sum_{j \neq J(i)} \left(x^{ij}_k - x^{iJ(i)}_k\right) w_{ij} + A_i x^{iJ(i)}_k - \lambda\Big(-\sum_{f \neq f^*(i,k)} \delta(x^i_{kf}) + \sum_{j \neq J(i)} \left(x^{ij}_k - x^{iJ(i)}_k\right)\delta(w_{ij})\Big) = 0 \tag{25}$$

If the entering variable is x^i_{kf}, then $x^i_{kf^*(i,k)}$ will become zero if

$$-\sum_{f \neq f^*(i,k)} x^i_{kf} - \lambda + \sum_{j \neq J(i)} \left(x^{ij}_k - x^{iJ(i)}_k\right) w_{ij} + A_i x^{iJ(i)}_k - \lambda\Big(-\sum_{f \neq f^*(i,k)} \delta(x^i_{kf}) + \sum_{j \neq J(i)} \left(x^{ij}_k - x^{iJ(i)}_k\right)\delta(w_{ij})\Big) = 0 \tag{26}$$

If the entering variable is $w_{ij'}$ then $x^i_{kf^*(i,k)}$ will become zero if

$$-\sum_{f \neq f^*(i,k)} x^i_{fk} + \sum_{j \neq J(i)} \left(x^{ij}_k - x^{iJ(i)}_k\right) w_{ij} + \left(x^{ij'}_k - x^{iJ(i)}_k\right)\lambda + A_i x^{iJ(i)}_k - \lambda\Big(-\sum_{f \neq f^*(i,k)} \delta(x^i_{kf}) + \sum_{j \neq J(i)} \left(x^{ij}_k - x^{iJ(i)}_k\right)\delta(w_{ij})\Big) = 0 \tag{27}$$

If λ solved from Equations (25), (26), or (27) will be the smallest one, the key factory $f^*(i,k)$ needs to be changed and the RHS needs to be updated.

Implementation of the algorithm

The implementation of the above-outlined algorithm in the J software is using linear algebra subroutines contained in quadratic programming software Bqpd made by Prof. R. Fletcher, University of Dundee based on Fletcher (1996). These subroutines are used to pivot columns of the working basis matrix $\mathbf{B}$ and to compute $\mathbf{c}'_b\mathbf{B}^{-1}$ (Eq.19) and $\mathbf{B}^{-1}\mathbf{d}$ (Equation 20).

A feasible solution is obtained by using a temporary objective function which is sum of constraint rows below their lower bound – the sum of constraints which are above their upper bounds.

Without factories, the algorithm is almost the same as in JLP (Lappi 1992). The only difference is that in J, the nonbinding constraints are treated in the standard way using residual variables. In JLP, the rows corresponding to nonbinding constraints are deleted from the working basis matrix.

Most alfas and betas in the objective Row (1) and in the utility Constraints (2) are zeros. This is taken into account in the computations by accessing only nonzero elements. This makes the data structures complicated.

Downloaded by [Luke Natural Resources Institute Finland] at 04:06 01 October 2015

Downloaded by [Luke Natural Resources Institute Finland] at 04:06 01 October 2015

An example

Our example refers to forestry board district North Savo. The forest data are from National Forest Inventory. In the data, there are 4336 sample plots, and each sample plot is considered to present a treatment unit. Each treatment unit is described by the coordinates. Plot variables are multiplied by an expansion factor so that the total area can be considered to consist of these units. The treatment schedules are generated with the MELA simulator (Redsven et al. 2012) for five 10-year periods, allowing different thinning, final harvest, and regeneration options for each unit. The number of treatment schedules (all w_{ij}'s i.e. $\sum_{i=1}^{m} n_i$) is 2081902, i.e. on the average 480 schedules per unit. The harvests are assumed to be done at the midpoints of the periods.

There are eight sawmills in the area. Each sawmill is described by the coordinates, the sawmill capacity, and the factory price of saw logs. There could be different capacities and factory prices for different periods, but in this example, the same capacity and price applies for all periods. In this example, the saw log price is 67€/m^3 for all sawmills. There are three pulp mills. Each pulp mill is also described by its coordinates, the capacity, and pulp wood price. Here the pulp wood price is assumed to be 38€/m^3 for all pulp mills. Transportation cost is assumed to be 0.064€/m^3/km, and the distance between a unit and a factory is assumed to be 1.5 times the Euclidian distance computed from the coordinates. If exact distances (or transportation costs generally) between a unit and the factories are available, they can be loaded from a file. There are 10 transported items (five periods for saw logs, and five periods for pulp wood) and 11 factories, thus the number of x^i_{kf}-variables in the original problem formulation is 4336 × 10 × 11 = 476,960, but taking into account that saw logs are transported to sawmills and pulp wood is transported to pulp mills (which can be described by setting part of α's and β's to be zero), the effective number of x^i_{kf}-variables is 4336 × 5 × 11 = 238,480.

The objective function to be maximized is the net present value, which can be described as:

$$\sum_{t=1}^{T}\sum_{s=1}^{S}\sum_{i=1}^{m}(-R_t * c_{tsi} + R_t u_{ts})RL_{tsi} + \sum_{t=1}^{T}\sum_{p=1}^{P}\sum_{i=1}^{m}\left(-R_t * d_{tpi} + R_t v_{tp}\right)RP_{tpi} - \sum_{t=1}^{T} R_t W_t + R_{T+}ENPV_{T+} \tag{28}$$

where t is the period, T is number of periods, s is sawmill, S is number of sawmills, i is unit, m is number of units, R_t is the discounting factor for period t (interest rate is assumed to be 3%), c_{tis} is the transportation cost for saw logs in period t between unit i and sawmill s, u_{ts} is the saw log price at sawmill s for period t, RL_{tif} is the harvested saw log volume transported in period t from unit i to sawmill s, d_{tis} is the transportation cost for pulpwood in period t between unit i and pulp mill p, v_{ts} is the pulp wood price at pulp mill p for period t, RP_{tif} is the harvested pulp wood volume transported in period t from unit i to pulp mill p, W_t is total amount (i.e. sum over units) of expenditures (e.g. regeneration and harvesting costs) in period t, R_{T+} is the discounting factor at the end of the planning horizon and $ENPV_{T+}$ is the estimated total (i.e. sum over units) net present value at time T of net revenues after the planning period assessed using average stumpage prices and simulating standard treatments to the end of rotation, whose length is obtained from common recommendations, and adding the land value at the end of the rotation as a function of forest type and temperature sum.

Comparing to the general formulation (Equation. (1)–(7)), a y_{kf} variable is, e.g. $\sum_{i=1}^{m}(-R_t * c_{tsi} + R_t u_{ts})RL_{tsi}$ The part depending on the factory price u_{ts} (and not depending on the unit location) could be defined as a $\alpha_{tkf} x_{kf}$-term, but as discussed above, it is more efficient to combine unit location independent and location dependent terms together. Period t and variable RL together correspond to index k in y_{kf}, and s is one example of index f.

The capacity constraints for sawmills are $RL_{ts} \le C_{ts}, t = 1, \ldots, T, s = 1, \ldots, S$ and for pulp mills $RP_{tp} \le D_{tp}, t = 1, \ldots, T, p = 1, \ldots, P$.

J has a very compact (but a little complicated) way to describe the above kinds of objectives and capacity constraints. The example problem has also sustainability constraints, requiring that total harvested volume is nondecreasing, harvested saw log volume is nondecreasing, and $ENPV_{T+}$ is not smaller than the total initial net present value computed using average stumpage prices also during the planning period and using $ENPV_{T+}$ to evaluate incomes after the planning period. This net present value constraint is used to guarantee that the value of the forest does not decrease during the planning horizon. In this constraint, NPV computed using Equation (28) could also be used for the initial net present value, but computed this way, the initial net present value is more compatible with $ENPV_{T+}$.

The problem has 64 utility constraints (2). The number of area constraints (4) is 4336. The number of transportation constraints (7) is 4336 × 10 = 43,360. The solution of the problem took 2 minutes and 18 seconds using a standard desktop computer (Intel i5–3210 CPU at 2.5 GHz) Standard sustainability LP problems without factories take in the same data 10–20 seconds.

The printed solution gives the value of the objective function, the values of all utility constraint rows, their shadow prices, and the values of all forest variables in the data and their shadow prices (the shadow prices of constraints (3), see Lappi 1992). There are J functions which can be used to access nonzero w_{ij} variables and x_{kf}^{i} variables (timber variables transported to different factories) and write them to files. Also the shadow prices of the area constraints (4) can be accessed.

Discussion

One difference between DTRAN and J in their capabilities to handle transportation costs is that DTRAN has tools to analyze road networks, while in J, the transportation costs are either estimated from coordinates or they can be obtained from files which are generated with other software. We do not have plans to add tools to J to handle road networks. J is solving LP problems "exactly," while DTRAN is solving them approximately. DTRAN is producing integer solutions by selecting one schedule for each unit. J is splitting units among schedules, but J has functions to round the solution to nearest integer solution.

There are many kinds of optimization problems in forestry, dealing with transportation costs which cannot be analyzed within the presented formulation (see Rönnqvist 2003 for a review). Planning problems at the tactical or operative level need more specific problem formulations and solution methods.

When considering the use of the J software for timber management planning, the most demanding requirement is, as in the previous versions of J, that there must be a simulator which generates the treatment schedules. The generation of transportations costs can be based on the coordinates as in our example, or more sophisticated software can generate the costs. The integration of J to a decision support system can be done using files which are read or written by J.

Gunn (2009) has analyzed strategic aspects of forest management and supply chain in a way which is similar to the approach presented here. One technical difference is that Gunn (2009) is defining separate transportations costs for regions, while in our formulation, costs are defined separately for each stand. This makes our problem in principle larger, but hopefully, the developed solution algorithm can compensate this. The presented algorithm and J software may be of interest when analyzing the interaction of long-term harvest scheduling and factory locations and capacities. Possible applications may also be the analyses of energy wood production where transportations costs play a dominant role. There have not yet been case studies using the new factory properties of J. The algorithm requires quite complicated data structures; therefore, it probably would not be possible to implement the presented algorithm using any existing optimization software.

References

Bergseng E, Ask JA, Gobakken T, Framstad E, Hoen HF, Solberg B. 2012. Balancing biodiversity protection and economics in long term boreal forest management: combining quantitative modelling and subjective opinions. Forest Policy Econ. 15:12–21. Available from: http://www.sciencedirect.com/science/article/pii/S1389934111001857

Camenson D. 1996. Spectrum: an analytical tool for building natural resource management tools. In: Päivinen R, Roihuvuo L, Siitonen M, editors. Large scale forestry scenario models: experience and requirements, vol. 5. Helsinki: EFI Proceedings; p. 133–142.

Dantzig GB, VanSlyke RM. 1967. Generalized upper bounding techniques. J Compt Sys Sci. 1:213–226.

Dykstra DP. 1984. Mathematical programming for natural resource management. New York: McGraw-Hill; 318 p.

Gunn E. 2009. Some perspectives on strategic forest management models and the forest products supply chain. INFOR: Inf Sys Oper Res. 47:261–272.

Fletcher R. 1996. Dense factors of Sparse matrices. Dundee Numerical Analysis Report NA/170

Hoen H. 1996. Forestry scenario modelling for economic analysis – experiences using GAYA-JLP model. In: Päivinen R, Roihuvuo L, Siitonen M, editors. Large scale forestry scenario models: experience and requirements, vol. 5. Helsinki: EFI Proceedings; p. 78–88.

Hoganson HM, Kapple DC. 1991. DTRAN version 1.0. A multi-market timber supply model. Users' guide. Minneapolis: University of Minnesota Department of Forest Resources Staff Series Paper No. 82.

Hoganson HM, Rose DW. 1984. A simulation approach for optimal timber management scheduling. Forest Sci. 30:220–238.

Kangas A, Rasinmäki J, editors. 2008. SIMO – Adaptive simulation and optimization for forest management planning. 31 p. Helsinki: University of Helsinki Department of Forest Resource Management Publications 41; 40 p.

Lappi J. 1992. JLP – a linear programming package for management planning. Finnish Forest Research Institute Research papers; 414, 134 p. Available from: http://www.metla.fi/products/J/

Lappi J, Lempinen R. 2013. J user's guide. Version 2.0. Available from: http://mela2.metla.fi/mela/j/manuals/J2.0_userguide.pdf

Luenberger DG, Ye Y. 2008. Introduction to linear and nonlinear programming. 3rd ed. New York: Springer; 546 p.

Melo MT, Nickel S, Saldanha-Da-Gama F. 2009. Facility location and supply chain management – A review. Eur J Oper Res. 196:401–412.

Rasinmäki J, Kalliovirta J, Mäkinen A. 2009. SIMO: an adaptable simulation framework for multiscale forest resource data. Comp Elect Agr. 66:76–84.

Redsven V, Hirvelä H, Härkönen K, Salminen O, Siitonen M. 2012. MELA2012 Reference Manual. Vantaa: The Finnish Forest Research Institute; 666 p. ISBN: 978-951-40-2394-1 (PDF). Available from: http://mela2.metla.fi/mela/julkaisut/oppaat/mela2012.pdf

Rönnqvist M. 2003. Optimization in forestry. Math Program Ser B. 97:267–284.

Turner BJ, Church R. 1995. Review of the use of the FORPLAN (FORest PLANning) model. Victoria: Department of Conservation and Natural Resources. ISBN: 0 7306 4011 6.

Vila D, Martel A, Beauregard R. 2006. Designing logistics networks in divergent process industries: a methodology and its application to the lumber industry. Inter J Prod Econ. 102:358–378.

Downloaded by [Luke Natural Resources Institute Finland] at 04:06 01 October 2015