

Article

Decision Support Systems (DSS) Optimal—A Case Study from the Czech Republic

Robert Marušák *, Jan Kašpar and Petr Vopěnka

Faculty of Forestry and Wood Sciences, Czech University of Life Sciences Prague, Kamýcká 129, Praha 6 Suchbátka 165 21, Czech Republic; E-Mails: kasparj@fld.czu.cz (J.K.); vopenka@fld.czu.cz (P.V.)

* Author to whom correspondence should be addressed; E-Mail: marusak@fld.czu.cz; Tel.: +420-22438-3712.

Academic Editor: Harald Vacik

Received: 7 November 2014 / Accepted: 24 December 2014 / Published: 8 January 2015

Abstract: Forest managers have traditionally planned harvests using their expert knowledge. This applies mainly to the spatial distribution of harvest units. The amount of timber to be harvested is regulated by market demand. In addition to forest managers' expert knowledge, there is a set of rules, which can be automatized. Computerized harvest planning will lead not only to saving time of forest managers, but it will also enable them to explore various scenarios in a matter of minutes. We introduce *Optimal*—GIS tool for spatial and temporal decisions of harvest scheduling. *Optimal* allows creating new harvest units, which reflects the forestry act and/or forest managers' requirements. *Optimal* includes necessary tools for automatic controlling of harvest unit parameters. It allows alternative harvest scheduling, while taking into account different constraints. *Optimal* is a decision Support System designed and applied for clear-cut and shelterwood silvicultural systems with respect to the environmental and economic constraints.

Keywords: harvest scheduling; model; decision support systems (DSS)

1. Introduction

Since the 18th century, a number of different methods have been developed for organizing forest production with the even flow requirements in Europe [1]. The concept of the ideal normal even-aged

forest [2] is used in most cases. However, its application in practical forestry is problematic because of nature disturbances, which will potentially increase due to the climate change [3]. Even so, the timber indicators used in many central European countries are derived from the concept of the normal even-aged forest. However, a regulated forest with a balanced and regulated age-class distribution is not only difficult to achieve, but also undesirable for long-term forest stability. The disadvantages associated with the use of these indicators in Central Europe have been outlined in previous publications [4–6].

For the reasons mentioned, there is an increasing need to analyze the development of spatial structure because without the spatial aspect, it is impossible to maintain environmental, social and other aspects of forest management [7]. Methods of operational research in conjunction with modern information technology and geographic information systems (GIS) can be used to create endogenous scheduling model [8] and a new type of forest management plans that consider spatial structure. What makes spatial forest-management plans different from conventional plans is the proposal of size, shape and position of the forest harvest units in the forest management area. Decision support systems (DSS) can be helpful tools for forest managers to take into account all spatial restrictions of management in forests.

There are two timber-harvesting indicators for harvest scheduling in Czech Republic that are implemented in Czech forestry legislation. These express the maximum possible final cut. One of them is known as the cutting percentage (hereafter referred to as CP) and it determines the percentage of harvest for each mature age class depending on rotation and regeneration period. According to this, CP ranges from 4% to 100%. The final cut by CP is calculated by multiplying the CP value (4%–100%) by the volume of age class [6,9]. These percentages used for CP were derived from the normal forest, as described in [2]. In addition, the CP indicator is static, incorporating planning for one decade only, without the option to account for harvesting possibilities over a longer time period and does not consider the spatial possibilities of harvesting. This results in strongly uneven, decadal harvests for the whole forest management area (FMA). Important social and political changes after 1989 in Czech Republic lead to new harvest scheduling possibilities. Unfortunately, there are few previous studies detailing new approaches of spatial harvest scheduling applied in Czech and Slovak Republic [4,10] as there are laws restricting clear-cut management systems. It is necessary to analyze the number of existing harvest approaches and models to find new techniques that comply with all social and political requirements.

Initially, it is important to analyze which type of harvest scheduling endogenous model is suitable for the conditions of Czech Republic. The predefined harvest units, expressed in unit-restricted models (URM), can cause large differences in harvested volume between planning periods [11] and low values of objective function [12]. Although the area restricted models (ARM) [12–15] appears more suitable for these reasons, their application for clear cut management system for the conditions of Czech Republic is problematic, as not only is the area of clear cut limited, but also the shape (width and length) of clear cut is limited.

It is also necessary to define scheduling goals. The main goal is to maximize wood production in commercial forests. This goal is often expressed by the maximum net present value of harvested volume (see for example [12,16,17]). An alternative way to express wood production maximization is simply by maximizing harvested volume. This is the simplest way to formulate the objective however harvest scheduling models with this type of objective formulation do not include the uncertainty of fluctuations in the timber markets which is the one of the most prominent sources of risk and uncertainty in forest

management [18]. However, approaches exist that can incorporate windows of uncertainty into the optimization model [19,20].

There are a number of other management goals or criteria that can be included in the models and as a consequence forest planning is very complex. In addition to the economic issue, there are two more particular issues; environmental and social [21]. The environmental issue (including biodiversity and nature conservation, carbon sequestrations, *etc.*) has already been included in many previous models (see for example [2,7,17,22,23] and references therein). However, there are fewer studies about social issues such as recreational function [24]. Further, many different law restrictions exist which are extremely variable from country to country.

Forest management is a complex problem as there are a number of management goals, requirements and restrictions to comply to. As a consequence there is the need to develop new decision support systems (DSS) which can be much more helpful in aiding forest manager in making the best decisions. A good overview and description of most of the available decision support systems in the world is presented on the website ForestDSS CoP [25].

The goal of this paper is to present the basic version of DSS *Optimal* (hereafter referred to as *Optimal*) for spatial harvest scheduling and its advantages compared to CP timber indicator, which is still used in the Czech Republic. Some of possibilities of DSS *Optimal* will be presented and discussed for the case of real forest management area in the Czech Republic.

2. Material and Methods

2.1. DSS *Optimal*

Optimal is intended to compare with conventional timber indicators that exist in Czech Republic for example, CP. From this reason, only one objective function (harvested volume maximization) is included in the model as the harvest cost and other objectives cannot be included in the CP calculation.

There are different characteristics which can be used for DSS description [26]: Temporal scale; Spatial context; Spatial scale; Decision-making dimension; Objective dimension; and goods and services dimension. *Optimal* is useful for tactical forest planning [2]. There are quite strict restrictions for so called green-up constraints in the Czech Republic (and other central European countries) and it is not possible to schedule harvesting without considering neighborhood relations. *Optimal* is developed for stand level harvesting and its purpose is for a single decision-maker as *Optimal* has a single objective dimension. Finally, *Optimal* is oriented for market wood production.

Optimal is an ArcGIS extension, which is being developed to support clear-cut and shelterwood forest management in central Europe. According to Borges *et al.* [26], *Optimal* has all the main components of DSS including: the database management system—it is possible to use all databases supported by ArcGIS software; the model base management system—the growth model based on [27] was used for growth prediction; and methods base management system—integer optimization model is used for spatial harvest schedule. This model works with the harvested volume maximization such as the objective of the model. Additionally, there are adjacency constraints, which are derived according to [28] and the constraints regarding harvest flow over the planning horizon. The harvest flow constraints are

defined by different harvest volumes between consecutive periods, such as [22]. The logical structure of *Optimal* is presented in Figure 1. For the detailed mathematical formulation of the model, see Appendix.

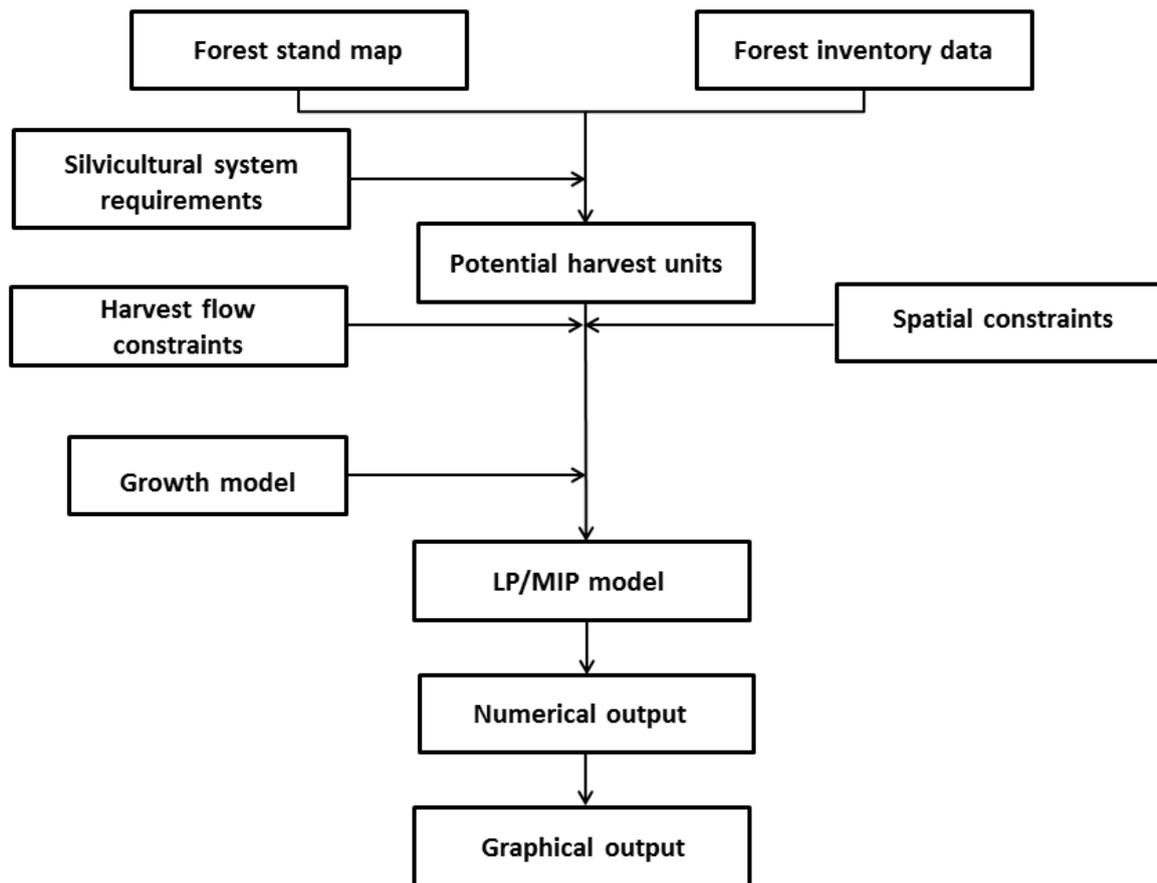


Figure 1. The logical structure of *Optimal*.

The description of *Optimal* can be divided into two parts: GIS and Solver. The GIS part is fully processed in ArcGIS framework, some original ArcGIS functions are used and functions have been added. The GIS part of *Optimal* provides the loading and displays the forest management data, sets the parameters of optimization, edit the new harvest units and creates the adjacency matrix. After simple installation, the toolbar (Figure 2) can be displayed in the ArcGIS framework. This toolbar includes all necessary components for harvest scheduling by *Optimal*.

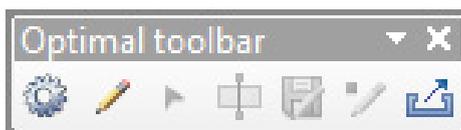


Figure 2. The *Optimal* toolbar.

The first button is for the parameter setting, such as minimum and maximum harvest units’ width and area. The next five buttons are for the cutting and editing process. The set parameters of harvest units are automatically checked by the system. The automatic checking of new harvest units is shown in the Figure 3 for the example of too narrow harvest unit.

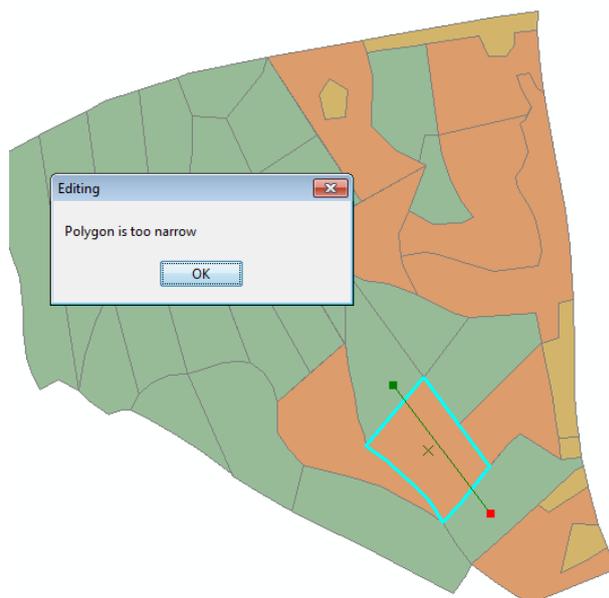


Figure 3. An example of the automatic checking process.

The last button is to start the Solver. The optimization package Gurobi [29] is used as a solver for the defined optimization model. It is directly linked to *Optimal* through Java API. The Java code for transferring the data from the GIS part of *Optimal* to solver and back to the results display is presented in Appendix.

The results of the harvest optimization model are displayed for users in three ways. First, after finishing the optimization, the summary results are displayed (Figure 4).

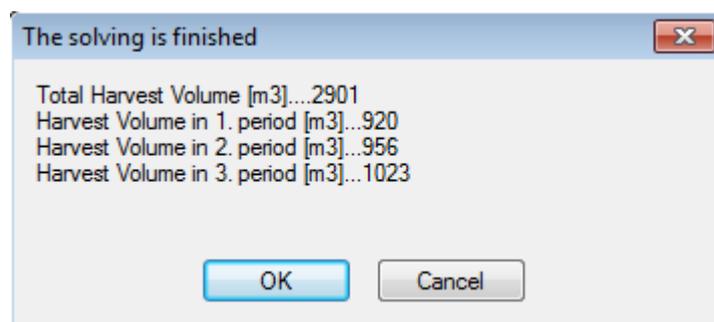


Figure 4. The summary results.

Secondly, the results are displayed in the graphical format as shown in Results and Discussion part (Figure 9) and finally, the new attribute column is added to the attribute table of input shapefiles during the solution. Each harvest units is identified by a number (from 0 to n), indicating the period in which the unit is designed for harvesting. 0 means not harvest and 1 means to harvest in the 1st period, etc.

The main difference between *Optimal* and other available DSS, is the possibility to create new harvest units and edit them according to *a priori* set of parameters. It is an important part of *Optimal* because it is necessary to divide each forest stand into the harvest units, which are exactly defined by the Czech Forestry Act. The maximal area of a harvest unit is one hectare in most cases, two hectares in special cases, and the maximal width of one harvest unit has to be equal or less than double of the mean height

of surrounding forest stands. The *Optimal* includes necessary tools for automatic controlling of mentioned parameters. Further, the potential harvest unit can be located in the minimal distance of one mean height of surrounding forest stand from any unit harvested in the same planning period. *Optimal* allows alternative scheduling while taking into account different constraints.

2.2. Case Study

Optimal is presented for the case of real FMA (Figure 5). The presented FMA has 494.8 hectares with Norway spruce (*Picea abies* L. Karst.) the predominant tree species in 342 of 425 forest stands (456 hectares; 92% from total area of FMA). The presented FMA is an average FMA of the Czech Republic in terms of its area, altitude (around 400 m), geology, and soil properties. For this FMA, alternative harvest schedule for 3 planning periods (each 10 years long) was created. This length of one planning period corresponds with traditional planning horizons used in the Czech Republic.

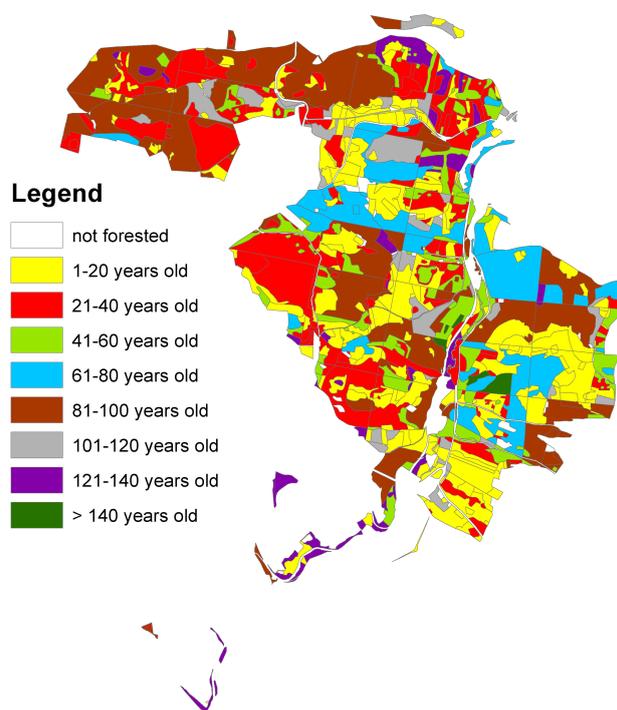


Figure 5. The forest stands map with colored age classes (20 years age class intervals).

A database of FMA includes information about each forest stand such as area, tree species composition, site index, age, regeneration period and rotation age. The average rotation age is 110 years and regeneration period is 30 years. All forest stands with an age older than their rotation age minus $\frac{1}{2}$ the length of the regeneration periods were selected from the database. These forest stands could be potentially harvested in the 1st planning period. The forest stands for harvesting in 2nd and 3rd periods were selected similarly (age + 10 years; resp. + 20 years is higher than the rotation age minus $\frac{1}{2}$ of length of the regeneration period).

It is obvious from Figure 6 that the real areas of 10-years age classes are quite different from the regulated areas of age classes, which is derived according to the mean rotation and regeneration period

for the whole FMA [9]. It could be expected that allowable cut indicators based on the model of normal forest cannot provide a good harvest schedule.

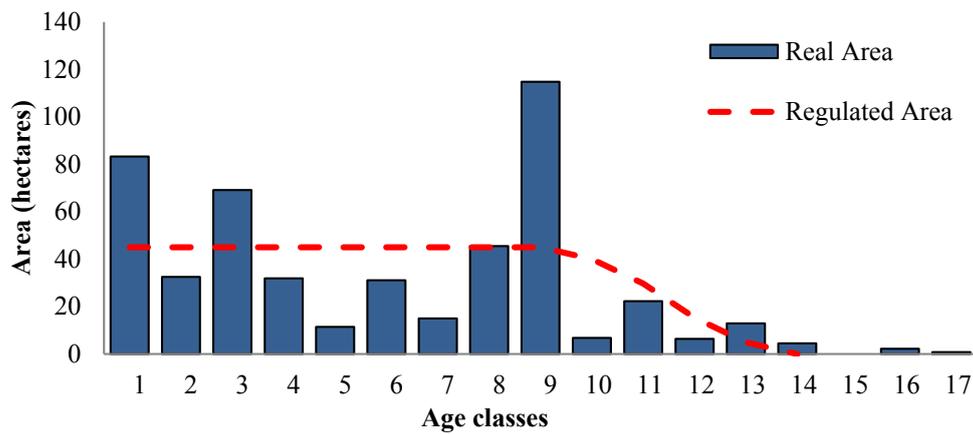


Figure 6. The real and regulated areas of 10-years age classes for FMA (forest management area).

All the selected forest stands were divided into the potential harvest units by using the Editing Tool of *Optimal* (Figure 7). The maximal width of clear-cut (50 m), the minimal width of clear cut (25 m) and the maximal area of clear cut (1 hectare) were defined.

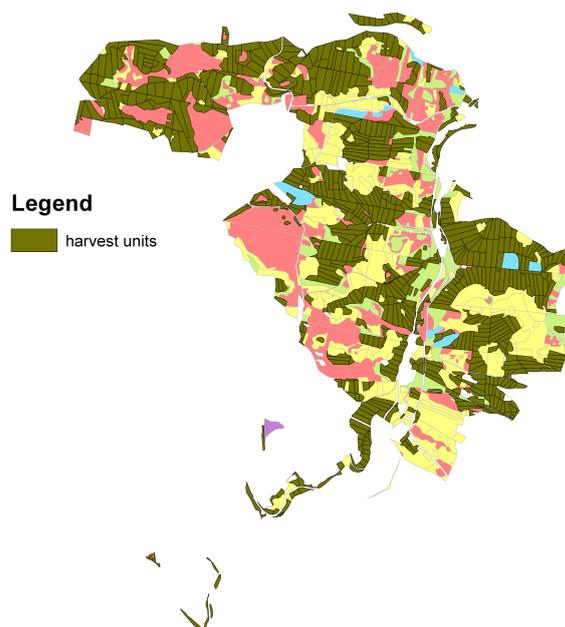


Figure 7. The potential harvest units for 3 planning periods.

The spatial harvest-scheduling tool *Optimal* includes adjacency constraints and harvest-flow constraints. Further, it is possible to choose between Moore and Neumann neighborhoods [10] and set the allowable gap tolerance accordingly.

Different variants of scheduling problem for presented FMA were calculated to compare with the conventional approach, CP. The variant AdC includes only adjacency constraints. The variant HfC

includes only harvest-flow constraints and the variant ComP includes both the adjacency and harvest-flow constraints. The sub-variants of variant HfC were calculated according to the harvest flow (from 1% to 60%) between the consecutive periods. The variants ComP and AdC were divided into two groups according to the Neumann (assigned as ComP-N and AdC-N) and Moore neighborhoods (ComP-M and AdC-M). As computational time is a limiting factor when using the optimization model, gap tolerance was also analyzed. Different gap tolerances (1.00×10^{-4} , 1.00×10^{-3} , 1.00×10^{-2} , and 1.00×10^{-1}) were used to calculate sub-variants of ComP-N; AdC-N; ComP-M and AdC-M. For variants ComP-N and ComP-M, different harvest flow (10%, 20% and 30%) was also calculated.

3. Results and Discussion

The 777 harvest units for clear-cut management system were edited for the purpose of the presented case study. The mean area of one harvest unit is 0.30 hectares, the maximum is 1.52 hectares and the minimum is 0.1 hectares. The average number of neighbors is 2 in the case of Neumann neighborhood and 5 in the case of Moore neighborhood. The planning horizon was defined for three periods at ten-year intervals.

The results of all evaluated variants and conventional scheduling problems are presented in Tables 1–4. For the resulting age structure dominated by the 8th and 9th age classes, the final cuts are rising in each period. The percentage differences between consecutive periods are small despite that CP does not include any type of harvest-flow constraints (increase of 2.4% in the second period compared to the first period; and 13.3% in the third period compared to the second period). However, the spatial aspect is missing, which means it could not be implemented in practice as adjacency constraints will not allow cutting more than presented in Tables 2 and 3.

Table 1. The scheduled total amount of harvest and individual harvests for three planning periods for CP (cutting percentage).

Total Harvested Amount (m ³)	Harvest in 1st Period (m ³)	Harvest in 2nd Period (m ³)	Harvest in 3rd Period (m ³)
102,032	31,824	32,598	37,610

It is obvious that when incorporating any type of adjacency constraints into the model, it is almost impossible to achieve the amount of harvest wood suggested by CP. Only sub-variants of ComP-N (Table 3) with 20% and 30% harvest flow are higher than the harvested amount suggested by CP. For the case of ComP-N variants, the total harvested area can be greater than the suggested CP as the total harvested units can be distributed over the FMA. Similar results were achieved by [30]. Although it is not possible to use the Neumann neighborhood constraints in any clear-cut harvest-scheduling model in the Czech Republic because of the Czech Forestry Act, they could be applied in the case of shelterwood management system.

It is shown that the elapsed time of solver depends on the gap tolerance. These results were expected because when gap tolerance is high then the solution is reached sooner. It seems that the elapsed time of solver also depends on the percentage of harvest flow for the same reason, as there are more possibilities how the harvested units can be distributed over the FMA. There are studies comparing the time and computational efficiency of different type of adjacency constraints [31,32], but unfortunately there

were no studies, which could confirm this fact. Only a small number of authors (e.g., [22]) indicate that it is more time-consuming to solve the problem with the harvest flow constraints than with the adjacency constraints.

Table 2. The harvest schedule for three planning periods for different sub-variants of ComP-M according to harvest flow and gap tolerance.

Gap tolerance	Resulted gap	Total harvested amount (m ³)	Harvest in 1st period (m ³)	Harvest in 2nd period (m ³)	Harvest in 3rd period (m ³)	Elapsed time of solver (s)	Elapsed time of <i>Optimal</i> (s)
The harvest flow difference 10%							
1.00×10^{-4}	0.0022%	80,157	24,217	26,638	29,302	0.45	242
1.00×10^{-3}	0.0517%	80,157	24,217	26,638	29,302	0.40	230
1.00×10^{-2}	0.7606%	79,701	24,101	26,450	29,150	0.34	230
1.00×10^{-1}	7.8183%	76,182	23,016	25,317	27,849	0.25	233
The harvest flow difference 20%							
1.00×10^{-4}	0.0036%	88,051	24,183	29,031	34,837	1.57	236
1.00×10^{-3}	0.0562%	88,029	24,193	29,018	34,818	0.55	235
1.00×10^{-2}	0.5092%	87,910	24,154	28,980	34,776	0.41	245
1.00×10^{-1}	9.1537%	82,753	22,737	27,280	32,736	0.27	232
The harvest flow difference 30%							
1.00×10^{-4}	0.0007%	91,216	22,867	29,722	38,627	2.24	244
1.00×10^{-3}	0.0530%	91,216	22,870	29,724	38,622	1.97	239
1.00×10^{-2}	0.6636%	91,108	22,877	29,674	38,557	0.50	240
1.00×10^{-1}	9.6607%	87,141	21,882	28,375	36,884	0.21	234

Table 3. The harvest schedule for three planning periods for different sub-variants of ComP-N according to gap tolerance.

Gap tolerance	Resulted gap	Total harvested amount (m ³)	Harvest in 1st period (m ³)	Harvest in 2nd period (m ³)	Harvest in 3rd period (m ³)	Elapsed time of solver (s)	Elapsed time of <i>Optimal</i> (s)
The harvest flow difference 10%							
1.00×10^{-4}	0.0089%	94,782	28,635	31,499	34,648	0.42	2557
1.00×10^{-3}	0.0298%	94,778	28,635	31,497	34,646	0.36	2470
1.00×10^{-2}	0.8973%	94,309	28,494	31,341	34,474	0.27	2677
1.00×10^{-1}	8.9499%	92,202	27,856	30,641	33,705	0.18	2531
The harvest flow difference 20%							
1.00×10^{-4}	0.0097%	104,231	28,635	34,362	41,234	0.58	2581
1.00×10^{-3}	0.0097%	104,231	28,635	34,362	41,234	0.59	2624
1.00×10^{-2}	0.8624%	103,384	28,404	34,082	40,898	0.49	2560
1.00×10^{-1}	5.4827%	99,665	27,382	32,856	39,427	0.28	2591
The harvest flow difference 30%							
1.00×10^{-4}	0.0015%	108,675	27,239	35,408	46,028	7.68	2613
1.00×10^{-3}	0.0948%	108,675	27,239	35,408	46,028	3.76	2634
1.00×10^{-2}	0.5647%	108,436	27,188	35,330	45,918	0.53	2547
1.00×10^{-1}	7.9413%	103,564	26,112	33,676	43,776	0.18	2841

Table 4. The harvest schedule for HfC (harvest-flow constraints) variant for 10%, 20% and 30% harvest flow and 1.00×10^{-4} gap tolerance.

Harvest Flow Difference	Total Harvested Volume (m ³)	Harvest in 1st Period (m ³)	Harvest in 2nd Period (m ³)	Harvest in 3rd Period (m ³)
10%	136,326	41,190	45,303	49,833
20%	136,827	37,612	45,098	54,117
30%	137,260	34,440	44,705	58,115

The time differences between sub-variants are not important from practical point of view. But, there are more important time differences between the sub-variants of ComP-M and sub-variants of ComP-N (Tables 2 and 3). The elapsed time needed for the whole process of creating the adjacency matrix under Neumann constraints, forming and solving the model, is approximately 10 times longer. This is connected with the actual algorithms to search for neighbors and it can be marked as a critical point of *Optimal*. The results confirm that increased computing speeds and improved commercial solvers and accepting unlimited number of constraints by solvers of course enable the ability to solve increasingly larger real problems [12]. However, there are so many factors, such as spatial structure, planning horizon length, type of harvest scheduling model and also the age structure of forest [31], which affect the solution time and efficiency.

The Table 4 presents harvest schedule for HfC variant for 10%, 20% and 30% harvest flow and 1.00×10^{-4} gap tolerance. These results are unrealistic because necessary adjacency constraints are missing; however the results show that adjacency constraints (spatial structure respectively) have significant effect on the total harvested volume. The total harvested volume suggested by different HfC sub-variants is from 50% to 70% higher than comparable total harvested volume suggested by different ComP-M sub-variants, which are the most realistic given the conditions of the Czech Republic. Furthermore, it seems that adjacency constraints have greater effect on the total harvested volume than harvest flow constraints. As is shown in Table 5, the total harvest volume over three planning periods does not increase as much with increasing harvest flow difference such as in the case of ComP-M and ComP-N variants. The significance of adjacency constraints is confirmed once more by the total harvest volume suggested by AdC-M variant and AdC-N variant (Table 5). The adjacency restrictions are less strict in the case of AdC-N variant and the total harvested volume is consequently higher than in the case of AdC-M variant.

Table 5. The harvest schedule for AdC-M and AdC-N variants for 1.00×10^{-4} gap tolerance.

	Total Harvested Amount (m ³)	Harvest in 1st Period (m ³)	Harvest in 2nd Period (m ³)	Harvest in 3rd Period (m ³)
AdC-M	92,806	12,083	29,799	50,204
AdC-N	110,200	8,111	34,568	67,521

The presented options of harvest flow constraints could result in lower remaining total FMA volume at the end of the planning horizon because a limit, not to harvest all the harvest units such as valuation of ending inventory stand volume, is missing [32]. This means that for the harvest flow constraints presented, harvest sustainability is not guaranteed for a longer planning horizon. Harvest flow constraints need to be revised in the optimization models, or more planning periods should be scheduled. In addition,

when adjacency constraints are included, there are many harvest units without harvesting and it is possible to identify them for harvesting in subsequent, potential planning periods. Further, the next potential harvest units should be created for the next potential planning periods because of their different age, rotation age and regeneration period.

Another possibility on how DSS *Optimal* can be beneficial and helpful for decision-making in forest management is presented in Figure 8. It shows that with increasing harvest flow, the total harvested volume also increases. However, this progress is quite small, from 25% of harvest flow, and the change in total harvested volume is insignificant from a practical point of view. Forest managers should choose between perfect harvest flow and harvested volume maximization in the real planning situation. However, with the presented information, forest managers can expect that the next increasing harvest flow has no effect on harvested volume maximization.

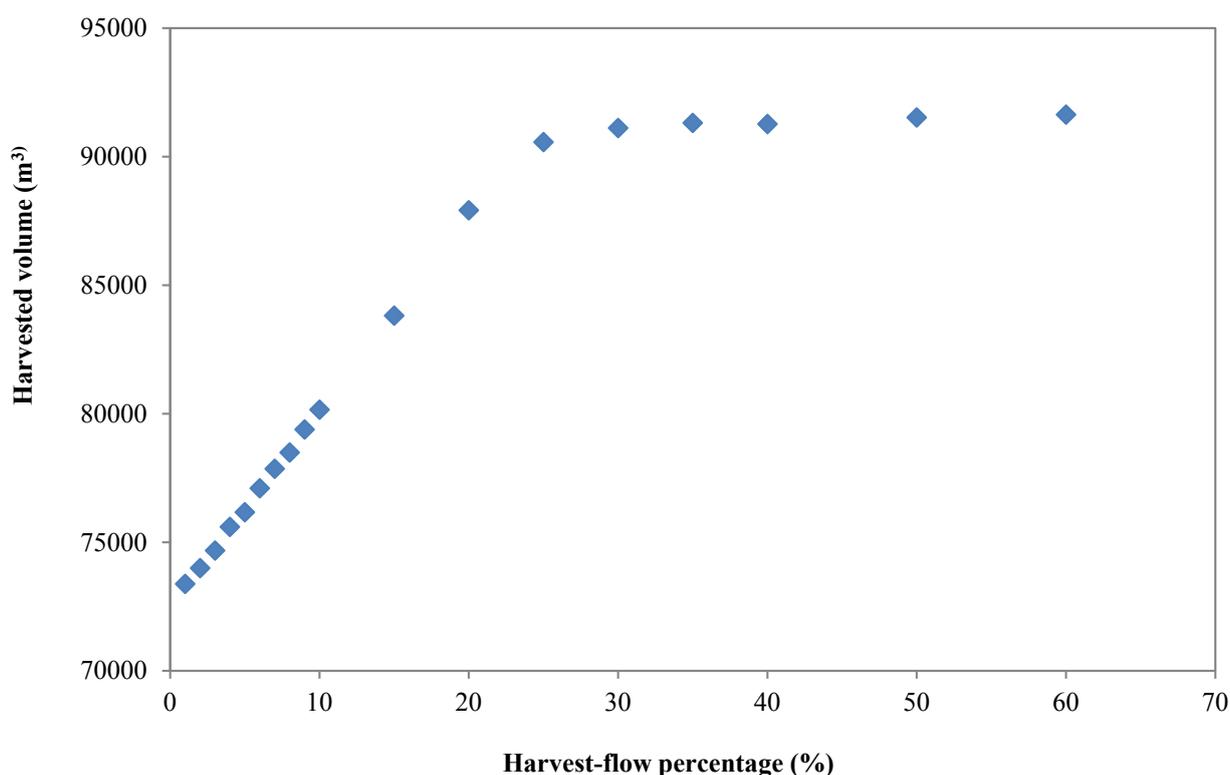


Figure 8. The relations between harvest flow percentage and total harvested volume for ComP-M variant and 1.00×10^{-4} gap tolerance.

The graphical outputs of DSS *Optimal* are also important for forest managers. Figure 9 presents spatial information about which harvest unit will be harvested and in which period. It is a big difference from the conventional way of harvest scheduling in the Czech Republic as the CP cannot provide spatial distribution of units to be harvested over the planning horizon. The choice of spatial location of harvest units is therefore subjectively determined by forest managers causing failure to achieve optimized harvests.

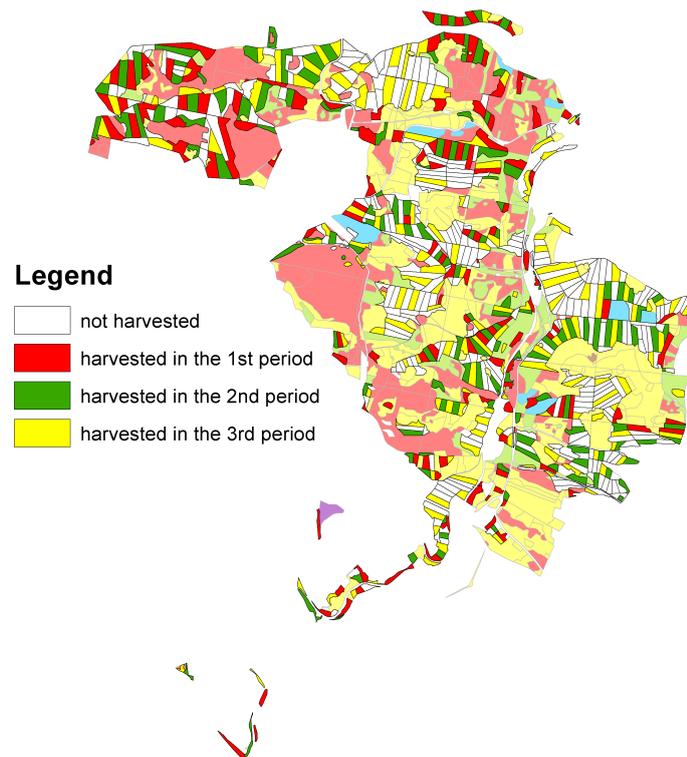


Figure 9. Graphical output of DSS (decision support systems) *Optimal* for ComP-M variant for 10% harvest flow difference and 1.00×10^{-4} gap tolerance.

4. Conclusions

The decision support system *Optimal* is a powerful system for harvest scheduling for the conditions of the Czech Republic. Unlike systems used in other part of the world, the decision support system *Optimal* includes tools for editing harvest units, which are restricted by Czech forestry law. It is presented that there are obvious advantages compared to allowable cut indicators currently used in the Czech Republic. This system is an alternative to conventional forest management plans used at the present time in the Czech Republic. Using *Optimal*, the forest managers will be able to change parameters and create various scenarios in a matter of minutes to find the best solution for them. As *Optimal* allows creating and checking strict spatial limits of harvest units, it is possible to apply *Optimal* not only in the Czech Republic but also in some other central European countries (e.g., Slovak Republic, Poland, *etc.*) where the forestry law requirements of limited harvest units are similar. *Optimal* uses Java SDK for ArcGIS desktop extensions. The future plan is to move from desktop solutions to server solutions to make it accessible to forest managers, to reduce initial costs and the need of software installation.

In the next step, the decision support system *Optimal* will be developed for other management systems and different non-productive functions such as biodiversity, water and soil protection, recreation, *etc.*

Acknowledgments

This research was supported by the project of the National Agency for Agriculture Research (No. QJ13202302) and the Internal Grant Agency of Faculty of Forestry and Wood Sciences Czech

University of Life Sciences in Prague (No. B0114). The authors would like to also thank Jennifer Clear for valuable proof reading.

Author Contributions

Róbert Marušák has set up the idea, has been directed the development of the DSS *Optimal* and created mathematical formulation of harvest scheduling model. Jan Kašpar has assisted with mathematical formulation of harvest scheduling model and has been responsible for programming in Java and transferring to Gurobi. Petr Vopěnka was responsible for creating ArcGIS extension.

Appendix

This appendix describes in detail the mathematical formulation of harvest scheduling model used in DSS *Optimal* and the way the model is transferred to the Gurobi software (Gurobi Optimization, Inc., Houston, TX, USA).

Mathematical Formulation

A FMA consisting of I harvest units, each one with the homogenous structure indexed by i . As this is a unit-restricted model [13] each binary variable in the model represents specifically one proposed harvest unit designed for harvesting or not over the P planning period.

Binary variables x is indexed by the harvest unit identifier; $i = 1, \dots, I$ and period $p = 1, \dots, P$.

$$x_{ip} = \begin{cases} 1 & \text{if the unit } i \text{ will be harvest in period } p \\ 0 & \text{in other cases} \end{cases} \quad (1)$$

One of these is that each unit can be cut just once per planned period. It can be generalized as:

$$\sum_{p=1}^P x_{ip} \leq 1 \quad \forall i = 1, 2, \dots, n \quad (2)$$

where n is the number of harvest units.

A harvest volume is allowed to increase or decrease by α from period to period. This can be expressed by the set of conditions regarding every pair of two consequential periods:

$$(1 + \alpha) \sum_{i=1}^I v_{ip} x_{ip} \leq \sum_{i=1}^I v_{i(p+1)} x_{i(p+1)}, \forall p = 1, \dots, P \quad (3)$$

$$(1 - \alpha) \sum_{i=1}^I v_{i(p-1)} x_{i(p-1)} \geq \sum_{i=1}^I v_{ip} x_{ip}, \forall p = 1, \dots, P \quad (4)$$

where α is the fractional difference permitted in the harvest level between two consequential periods p and v_{ip} is the volume of the wood in the unit i in the period p .

Conditions that originate in the spatial relations between the harvest units can be set down using analytic algorithm [28]:

$$M \cdot x \leq A \quad (5)$$

$$M = A + B \quad (6)$$

where A : adjacency matrix; B : diagonal matrix in which the i th diagonal element b_{ii} is defined by $b_{ii} = A_i \cdot I$ (A_i is i -th row vector of adjacency matrix A); M : modified adjacency matrix; x : control vector for control variables x_{ip} ; 1 is an $(n \times I)$ unit vector.

The objective function of optimization model is defined as:

$$\max Z = \sum_{i=1}^I \sum_{p=1}^P v_{ip} \cdot x_{ip} \quad (7)$$

Java programming code for the model transferring to Gurobi

```
package com.proforesters.solver;
/**
 * @author kaspar
 *
 */
import gurobi.GRB;
import gurobi.GRBEnv;
import gurobi.GRBException;
import gurobi.GRBLinExpr;
import gurobi.GRBModel;
import gurobi.GRBVar;

import com.esri.arcgis.geodatabase.IFeatureClass;
import com.proforesters.optimal.OptimalExtension;

public class ClearCutSystemSolver {
    public static double[] getSolution (int [][] matrix, int periodCount, int deviation, double [] []
    objectiveMatrix, int [] patches, IFeatureClass featureClass, int [][] timeHarvest, int gapTolerance)
    {

        double [] results = new double [periodCount * matrix.length];

        try {
            GRBEnv env = new GRBEnv("mip1.log");
            GRBModel model = new GRBModel(env);
            double gT = gapTolerance * 1000
            double doubleGapTolerance = gT/ 10000000;
            model.getEnv().set(GRB.DoubleParam.MIPGap,doubleGapTolerance);
            double decimalDeviation =((double)deviation)/100;
            int finalCountOfRow = matrix.length * periodCount + matrix.length
            +(2*periodCount - 2) + 1;
```

```

int [] finalConstraints = new int [finalCountOfRow];
int [] sumRow = new int [matrix.length];
int [] diagElem = new int [matrix.length];
for (int i = 0; i < matrix.length; i++) {
    for (int j = 0; j < matrix.length; j++) {
        sumRow [i] += (matrix [i][j]);}
for (int i = 0; i < matrix.length; i++) {
    diagElem [i] = (sumRow [i])/2;}
for (int i = 0; i < periodCount; i++) {
    for (int j = 0; j < matrix.length; j++) {
        finalConstraints [i*matrix.length+j] = diagElem [j];}
for (int i = matrix.length * periodCount; i < matrix.length * periodCount + 2 *
periodCount - 2; i++) {
    finalConstraints [i] = 0;}
for (int i = matrix.length * periodCount + 2 * periodCount - 2; i < finalCountOfRow
- 1; i++) {
    finalConstraints [i] = 1;}
int sumOfPatchesVector=0;
for (int i = 0; i < patches.length; i++) {
    sumOfPatchesVector += patches [i];}
for (int i = finalCountOfRow - 1; i < finalCountOfRow; i++) {
    finalConstraints [i] = sumOfPatchesVector;}
int n = matrix.length * periodCount;
GRBVar [] x = new GRBVar[n];
for (int i = 0; i < matrix.length * periodCount; i++) {
    String st = "x" + String.valueOf(i);
    x[i] = model.addVar(0.0, 1.0, 0.0, GRB.BINARY, st);}

model.update();

double [] objectiveVector = new double [periodCount * matrix.length];
for (int i = 0; i < periodCount; i++) {
    for (int j = 0; j < matrix.length; j++){
        objectiveVector [j+i*matrix.length] = objectiveMatrix [i][j];}

GRBLinExpr expr = new GRBLinExpr();

for (int i = 0; i < matrix.length * periodCount; i++) {
    expr.addTerm(objectiveVector[i], x[i]);}

model.setObjective(expr, GRB.MAXIMIZE);

int count = matrix.length*periodCount;

```

```

for(int i = 0; i < periodCount;i++){
    for (int j = 0; j < matrix.length; j++) {
        expr = new GRBLinExpr();
        for (int k = 0; k < matrix.length; k++) {
            expr.addTerm(matrix[j][k], x[(matrix.length * i) + k]);}
        String st = "c" + String.valueOf((matrix.length * i) + j);
        model.addConstr(expr, GRB.LESS_EQUAL, finalConstraints[j], st);} }

double [][] evenFlowMatrixUP = new double [periodCount-1][matrix.length *
periodCount];
for (int i = 0; i < periodCount - 1; i++) {
    for (int j = 0; j < matrix.length * periodCount; j++){
        evenFlowMatrixUP [i][j] = 0;}}
for (int i = 0; i < periodCount - 1; i++) {
    for (int j = 0; j < matrix.length; j++) {
        evenFlowMatrixUP [i][i*matrix.length+j] = -(1 + decimalDeviation) *
        objectiveMatrix [i][j];}}
for (int i = 1; i < periodCount; i++) {
    for (int j = 0; j < matrix.length; j++) {
        evenFlowMatrixUP [i - 1][i*matrix.length + j] = objectiveMatrix
        [i][j];}}
for (int i = matrix.length * periodCount; i < matrix.length * periodCount
+(periodCount - 1); i++) {
    expr = new GRBLinExpr();
    for (int j = 0; j < matrix.length * periodCount; j++) {
        expr.addTerm(evenFlowMatrixUP[i - (matrix.length *
periodCount)][j], x[j]);}
    String st = "c" + String.valueOf(i);
    model.addConstr(expr, GRB.LESS_EQUAL, finalConstraints[i], st);}

double [][] evenFlowMatrixLO = new double [periodCount-1][matrix.length *
periodCount];
for (int i = 0; i < periodCount - 1; i++) {
    for (int j = 0; j < matrix.length * periodCount; j++){
        evenFlowMatrixLO [i][j] = 0;}}
for (int i = 0; i < periodCount - 1; i++) {
    for (int j = 0; j < matrix.length; j++) {
        evenFlowMatrixLO [i][i*matrix.length + j] = -(1 - decimalDeviation)
        * objectiveMatrix [i][j];}}
for (int i = 1; i < periodCount; i++) {
    for (int j = 0; j < matrix.length; j++) {

```

```

        evenFlowMatrixLO [i - 1][i*matrix.length + j] = objectiveMatrix
        [i][j];}}
for (int i = matrix.length * periodCount +(periodCount - 1); i < matrix.length *
periodCount + 2*(periodCount - 1); i++) {
    expr = new GRBLinExpr();
    for (int j = 0; j < matrix.length * periodCount; j++) {
        expr.addTerm(evenFlowMatrixLO[i - (matrix.length * periodCount +
(periodCount - 1))][j], x[j]);}
    String st = "c" + String.valueOf(i);
    model.addConstr(expr, GRB.GREATER_EQUAL, finalConstraints[i], st);}
for (int k=0; k < matrix.length; k++){
    int [] oneForPeriodMatrix= new int [matrix.length * periodCount];
    for (int j = 0; j < matrix.length*periodCount ; j++){
        oneForPeriodMatrix [j] = 0;}
    for (int j = 0; j < periodCount ; j++){
        oneForPeriodMatrix [j * matrix.length + k] = 1;}
    expr = new GRBLinExpr();
    for(int j = 0; j < matrix.length * periodCount; j++){
        expr.addTerm (oneForPeriodMatrix[j], x[j]);}
    String st = "c" + String.valueOf(200000+k);
    model.addConstr(expr, GRB.LESS_EQUAL, 1, st);}

patchesVector [] = new int [matrix.length * periodCount];
for (int i = 0; i < patches.length; i++) {
    patchesVector [i] = patches [i];}
for (int i = finalCountOfRow - 1; i < finalCountOfRow; i++) {
    expr = new GRBLinExpr();
    for (int j = 0; j < matrix.length * periodCount; j++) {
        expr.addTerm(patchesVector [j], x[j]);}
    String st = "c" + String.valueOf(i);

model.addConstr(expr, GRB.LESS_EQUAL, finalConstraints[i], st);

for(int i = 1; i < periodCount;i++){
    for (int j = 0; j < matrix.length; j++) {
        if(timeHarvest[i][j]==1){
            for(int k = 0; k < i;k++){
                expr = new GRBLinExpr();
                expr.addTerm(timeHarvest[i][j], x[j + matrix.length * (k)]);
                String st = "a" + String.valueOf((matrix.length * i) + j);
                model.addConstr(expr, GRB.EQUAL, 0, st);}}}}

```

```

model.optimize();

for (int i = 0; i < matrix.length * periodCount; i++) {
    results [i] = x[i].get(GRB.DoubleAttr.X);}
    for (int i = 0; i < periodCount; i++){double suma = 0;for (int j = 0; j <
matrix.length; j++){
        suma += results[i*matrix.length+j] * objectiveMatrix[i][j] ;}}

OptimalExtension.setResults (featureClass, results, periodCount);
model.update();
model.write("test.lp");
model.dispose();
env.dispose();
return results;
} catch (GRBException e) {
    System.out.println("Error code: " + e.getErrorCode() + ". " +e.getMessage());
    for (int i = 0; i < matrix.length * periodCount; i++) {
        results [i] = 0;}
    return results;}}

```

Conflicts of Interest

The authors declare no conflict of interest.

References

1. Davis, L.S.; Johnson, K.N.; Bettinger, P.S.; Howard, T.E. *Forest Management: To Sustain Ecological, Economic, and Social Values*, 4th ed.; McGraw-Hill Higher Education: New York, NY, USA, 2001; p. 394.
2. Bettinger, P.; Boston, K.; Siry, J.P.; Grebner, D.L. *Forest Management and Planning*; Academic Press: New York, NY, USA, 2009; p. 329.
3. Hlásný, T.; Mátyás, C.; Seidl, R.; Kulla, L.; Merganičová, K.; Trombik, J.; Dobor, L.; Barcza, Z.; Konôpka, B. Climate change increases the drought risk in Central European forests: What are the options for adaptation. *For. J.* **2014**, *60*, 5–18.
4. Kašpar, J.; Marušák, R.; Sedmák, R. Spatial and non-spatial harvest scheduling *versus* conventional timber indicator in over-mature forests. *For. J.* **2014**, *60*, 81–87.
5. Marušák, R. Alternative harvest scheduling for final cut with respect to silvicultural requirements. *For. J.* **2007**, *53*, 117–127.
6. Marušák, R.; Yoshimoto, A. Comparative analysis on cutting possibilities derived from different allowable cut indicators in Slovakia. In *Formath Kobe*; Japan Society of Forest Planning Press: Tokyo, Japan, 2007; pp. 223–238.
7. Baskent, E.Z.; Keles, S. Spatial forest planning: A review. *Ecol. Mod.* **2005**, *188*, 145–173.

8. Kurtilla, M. The spatial structure of forests in the optimization calculations forest planning—A landscape ecological perspective. *For. Ecol. Manag.* **2001**, *142*, 129–142.
9. Priesol, A.; Polák, L. *Forest Management*; Priroda: Bratislava, Slovakia, 1991; p. 447. (In Slovak)
10. Konoshima, M.; Marušák, R.; Yoshimoto, A. Spatially constraints harvest scheduling for strip allocation under Moore and Neumann neighbourhood adjacency. *J. For. Sci.* **2011**, *57*, 70–77.
11. Richards, E.W.; Gunn, E.A. A model and tabu search method to optimize stand harvest and road construction schedules. *For. Sci.* **2000**, *46*, 188–203.
12. Crowe, K.; Nelson, J.; Boyland, M. Solving the area-restricted harvest-scheduling model using the branch and bound algorithm. *Can. J. For. Res.* **2003**, *33*, 1804–1814.
13. Crowe, K.; Nelson, J. An evaluation of the simulated annealing algorithm for solving the area-restricted harvest scheduling model against optimal benchmarks. *Can. J. For. Res.* **2005**, *35*, 2500–2509.
14. Murray, A.T. Spatial Restrictions in Harvest Scheduling. *For. Sci.* **1999**, *45*, 45–52.
15. Constantino, M.; Martins, I.; Borges, J.G. A new mixed-integer programming model for harvest scheduling subject to maximum area restrictions. *Oper. Res.* **2008**, *56*, 542–551.
16. Crowe, K.; Nelson, J. An indirect search algorithm for harvest-scheduling under adjacency constraints. *For. Sci.* **2003**, *49*, 1–11.
17. Öhman, K.; Lämås, T. Reducing forest fragmentation in long-term forest planning by using the shape index. *For. Ecol. Manag.* **2005**, *212*, 346–357.
18. Pasalodos-Tato, M.; Mäkinen, A.; Garcia-Gonzalo, J.; Borges, J.G.; Lämås, T.; Eriksson, L.O. Review. Assessing uncertainty and risk in forest planning and decision support systems: Review of classical methods and introduction of innovative approaches. *For. Syst.* **2013**, *22*, 282–303.
19. Palma, C.D.; Nelson, J.D. A robust optimization approach protected harvest scheduling against uncertainty. *Can. J. For. Res.* **2009**, *39*, 342–355.
20. Wei, R.; Murray, A.T. Spatial uncertainty in harvest scheduling. *Ann. Oper. Res.* **2012**, doi:10.1007/s10479-012-1178-2.
21. Diaz-Baltero, L.; Romero, C. Making forestry decisions with multiple criteria: A review and an assessment. *For. Ecol. Manag.* **2008**, *255*, 3222–3241.
22. Öhman, K.; Wikström, P. Incorporating aspects of habitat fragmentation into long-term forest planning using mixed integer programming. *For. Ecol. Manag.* **2008**, *255*, 440–446.
23. Yoshimoto, A.; Marušák, R. Evaluation of carbon sequestration and thinning regimes within the optimization framework for forest stand management. *Eur. J. For. Res.* **2007**, *126*, 315–329.
24. Yoshimoto, A.; Surovy, P.; Konoshima, M.; Surova, D. Optimal tourism management considering forest visual impression. In Proceedings of the International Symposium on: A New Era of Forest Management for Ecosystem Services, Seoul National University, Seoul, Korea, 28 June 2012.
25. Forest DSS Community of Practise. Available online: <http://www.forestdss.org> (accessed on 27 November 2014).
26. Borges, J.G.; Nordström, E.M.; Garcia-Gonzalo, J.; Hujala, T.; Trasobares, A. *Computer-Based Tools for Supporting Forest Management. The Experience and the Expertise World-Wide*; Department of Forest Resource Management-SLU: Umeå, Sweden, 2014; p. 507.

27. Černý, M.; Pařez, J.; Malík, Z. *Yields and Enumeration Tables for the Main Tree Species (Spruce, Pine, Beech, Oak)*; The Institute for Forest Ecosystem Research: Jílové u Prahy, Czech Republic, 1996; p. 245. (In Czech)
28. Yoshimoto, A.; Brodie, J. Comparative-analysis of algorithms to generate adjacency constraints. *Can. J. For. Res.* **1994**, *24*, 1277–1288.
29. Gurobi Optimizer Reference Manual, 5.6. Available online: <http://www.gurobi.com/documentation/5.6/reference-manual/refman.pdf> (accessed on 30 December 2014).
30. Konoshima, M.; Marušák, R.; Yoshimoto, A. Harvest scheduling with spatial aggregation for two and three strip cut system under shelterwood management. *J. For. Sci.* **2011**, *57*, 271–277.
31. McDill, M.; Braze, J. Comparing adjacency constraint formulations for randomly generated forest planning problems with four age-class distributions. *For. Sci.* **2000**, *46*, 423–436.
32. Hoganson, H.; Borges, J. Impacts of the time horizon for adjacency constraints in harvest scheduling. *For. Sci.* **2000**, *46*, 176–187.

© 2015 by the authors; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/4.0/>).