Computers and Electronics in Agriculture 113 (2015) 254–259

Contents lists available at ScienceDirect

Computers and Electronics in Agriculture

journal homepage: www.elsevier.com/locate/compag

Application note

GIS tool for optimization of forest harvest-scheduling

Petr Vopěnka*, Jan Kašpar, Robert Marušák

Dpt. of Forest Management, Faculty of Forestry and Wood Sciences, Czech University of Life Sciences in Prague, Prague, Czech Republic

ARTICLE INFO

Article history:
Received 15 September 2014
Received in revised form 22 February 2015
Accepted 4 March 2015
Available online 19 March 2015

Keywords:
Harvest scheduling
GIS
Optimization
Forest management

ABSTRACT

This article describes GIS tool (Optimal) for spatial and temporal optimization of forest harvests. Using Optimal, forest manager can create harvest units by editing polygons of forest stands in digital map. After the harvest units are created manually by the user, the adjacency matrix is automatically produced and passed to a solver module. The solver performs optimization using integer programming and returns spatial distribution of harvest units for each harvest period. User can set number of parameters, such as number and length of harvest periods, acceptable distances and areas of harvest units. The Optimal enables the forest managers to create and explore various scenarios and increase efficiency in forest harvest-scheduling.

© 2015 Elsevier B.V. All rights reserved.

1. Introduction

There are basically two main aspects of forest harvest-scheduling: Space and time. The forest spatial structure refers to the spatial arrangement of forest stands, harvest units or patches and interconnections among them (Baskent and Keles, 2005). The spatial structure plays important role in providing ecosystem services (Kurttila, 2001) and cannot be omitted in forest harvest scheduling. Temporal aspect is important for supplying good quality timber to the market according to market demand and at the same time preserving enough of it in the forest for the future.

The clear cut forest management system is commonly used in the Central Europe because of its cost efficiency. For preserving biodiversity and other non-timber forest products, the size and spatial relationship of the clear cuts is usually limited by law. The limitations can be expressed through four constraints: (1) The maximum area of the clear cut unit. The default is 1 ha, which is legal limit for clear cuts in the Czech Republic. (2) The minimum distance between the two clear cut units harvested in the same period, which is usually set equal to height of the forest stand. This would prevent the remaining forest stands from being vulnerable by wind. (3) The maximum width of the clear cut unit, which is usually set equal to legal limit of two heights of the forest stand. (4) Adjacency relationship, which is usually set to not to allow Queen's case (see below) as this is an official limit included in forestry legislation of the Czech Republic. Queen's case neighboring may be allowed in special cases where reconstruction of forest stands has to be done faster than usual. A neighboring clear cut unit can only be harvested when the area is regenerated to the point where it is stable forest stand again, so called green-up constraint (Bettinger et al., 2009). All these restrictions make harvest scheduling model computationally difficult to solve even for quite small forest management area. There are basically two possible modeling approaches to solve our problem, Area Restricted Models (ARMs) and Unit Restricted Models (URMs). It has been proved that ARMs have number of advantages over URMs (Richards and Gunn, 2000). For example higher values of total harvests or lower harvest flow percentages (Murray, 1999). However, because of the harvest unit shape restrictions we used URM modeling approach.

Today, most of the forest management plans can be designed only with the use of geographic information systems (Baskent and Keles, 2005). Over the last decade, there is increasing number of approaches, which deal with spatial aspects of harvest scheduling (Ohman and Eriksson, 2002; Ohman and Lamas, 2003, 2005; Baskent and Kelles, 2006). A decision support systems for spatial harvest optimization were developed, e.g. SNAP (Sessions and Sessions, 1988) or HEUREKA (Wikström et al., 2011). Other solutions used for this purpose like J-Software (Lappi and Lempinen, 2013) are rather development tools, not ready to use systems. These systems can optimize the spatial distribution of the harvest units, but lack the inbuilt editing and checking capabilities needed for construction of harvest units. Law restrictions for clear cut management system in number of countries are quite different, making it difficult to adopt single solution. The main objective of this paper is to develop a GIS tool to help forest managers with spatial harvest planning, including algorithms for creating harvesting units and estimating periodically harvesting flows.

---

* Corresponding author. Tel.: +420 22438 3796.
E-mail address: vopenka@fld.czu.cz (P. Vopěnka).

http://dx.doi.org/10.1016/j.compag.2015.03.001
0168-1699/© 2015 Elsevier B.V. All rights reserved.

2. Components of the framework

Optimal is an extension of proprietary geographic information system ArcGIS. It is combination of geographic information system (GIS) tool and mixed integer linear programming (MIP) solver. Optimal extension is designed for forest managers who have no understanding of MIP or any mathematics used in the model. However, basic knowledge of operating GIS is assumed. The basic structure of the software is schematically described in Fig. 1.

The extension uses Add-In concept introduced with ArcGIS version 10. The entire extension is packed into single file. When the file is double clicked it copies itself to appropriate location within ArcGIS installation directory. That way the extension is installed and ready to use. User starts the work by adding geographic data layer to ArcMap map composition. This can be either shapefile or ArcGIS geodatabase feature class. The geographic data layer should contain polygons of forest management units that are intended to be harvested. The data table of this layer must contain information about species and timber volume estimates per hectare for each forest stand. The volumes are automatically increased between periods using growth models designed for Czech main tree species (Černý et. al., 1996). There is no specific requirement as far as the names of the columns are concerned. User is required to select the columns, which contain the data. After the user selects the columns, the software performs validity check for numeric fields. The user sets up constraints for construction of harvest units in a special dialog box. The values depend on either legal restrictions or on desired shape of the harvest units. These constraints for the harvest units are: (1) minimal width, (2) maximal width, (3) minimal area and (4) maximal area.

When constraints for editing are set, user can start editing. System will automatically fill polygons with colours representing adherence to the constraints, e.g. whether the harvest unit is too large, or the harvest unit is too wide. That way the user has an overview, which polygons still needs to be edited, and which are already in line with the constraints that he chooses. The flow diagram is shown in Fig. 2.

In principle the editing is performed by cutting polygons of forest stands into smaller harvest units. Every time, just before the polygon is cut and resulting two new polygons saved to the

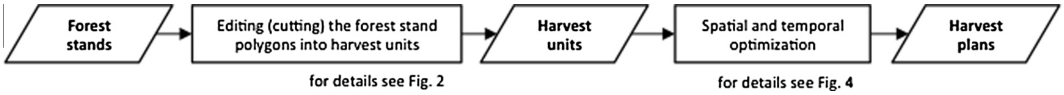

Fig. 1. Basic schema of the software components and workflow.

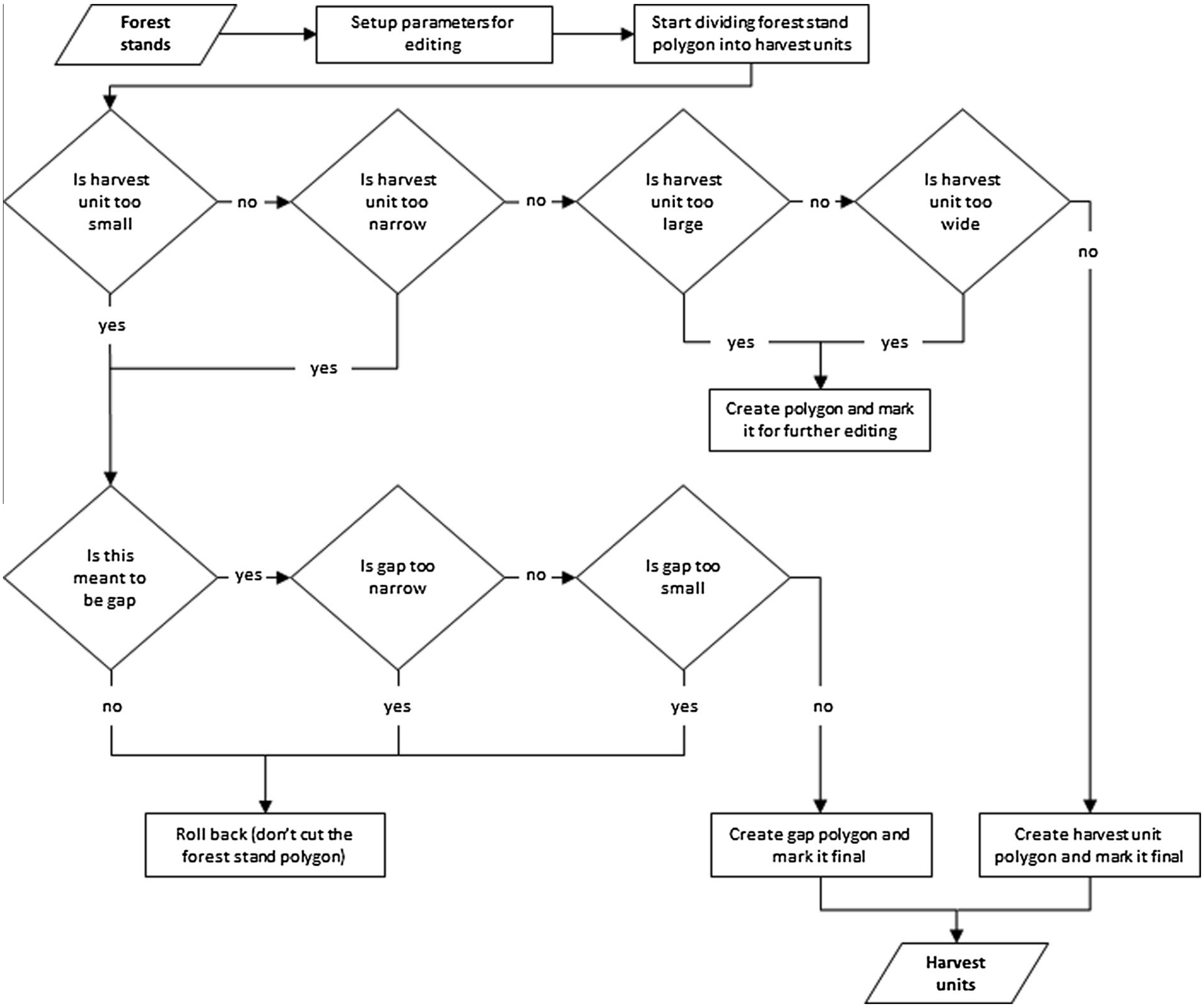

Fig. 2. Algorithm for manual editing forest stand polygons to produce harvest units.

database, the area and width of newly created polygons is calculated and checked against the constraints. If one of the new polygons is smaller or narrower than desired, user gets warning, the action is rolled back and new polygons are not created. If the new polygon is larger or wider then desired, then it is created anyway, but the polygon in the map is filled with appropriate colour, so that the user is notified that it still needs to be edited. We are checking the small and narrow polygons, because it is not efficient to harvest small forest patches.

The algorithm for checking width of the polygon uses inner buffers. Before the new polygon is saved, an attempt to create temporary inner buffer is made within it. The size of the inner buffer is equal to half of the minimal width set by the user. If it was impossible to create such a buffer, then it is clear that the width of the polygon is smaller than the desired minimal width. In such a case the new polygon is not created. The same principle is used for checking maximal width of the harvest unit polygon. However, in this case we first derive convex hull of harvest unit polygon and the inner buffer is constructed within the convex hull. If the system is able to create inner buffer using half of the set maximal width as a parameter for the buffer, then the convex hull of the polygon is considered too wide. The principle is illustrated in Fig. 3. If we set 25 m as a maximal width parameter for example, then the polygon would be considered too wide with (Fig. 3B) or even without (Fig. 3A) using the convex hull. However, setting the parameter to 50 m, the polygon is adhering to maximal width restriction with normal buffering (Fig. 3C), but not if we use the convex hull principle (Fig. 3D). Then the polygon would be still marked as if it needs further editing and would have to be cut to smaller polygons.

The harvest unit may be divided by narrow linear feature such as road, water stream, open area, or can be in a shape of crescent (see Fig. 3). Using convex hull eliminates these irregularly shaped harvest units to become large open areas. At this moment the convex hull principle is used as a default and cannot be changed by the user. In addition to harvest units, user can choose to include small gaps of significantly smaller size than harvest units. These artificial gaps are placed into large forest stands to create room for patches of either natural or artificial regeneration. User can setup two constrains for gaps: (1) minimal width and (2) maximal width. These gaps are treated differently not only when editing (cutting) the forest stands, but also when automatic optimization is performed. These gaps should be harvested in the first period so that there is a time for them to regenerate before the surrounding forest is harvested.

As soon as all the forest stands aimed to be harvested have been edited into harvest units or gaps, user sets up parameters for optimization. These are: (1) Maximal distance of neighbors. The harvest units that do not fall within the set distance from the source harvest unit are not considered to be neighboring harvest units. This makes it possible to include not only harvest units sharing border, but also harvest units that are within certain distance of source unit. (2) Choice whether the user wants to include only those polygons that are adjacent to each other so that they share a common boundary, so called Rook's case, or those that share either a common boundary or just a common vertex, so called Queen's case (Cho and Newman, 2005). Principles are similar to Moore and Neumann neighborhoods used in cellular automata (Balzter et al., 1998). This choice is only available if maximal distance of neighbors is set to zero. If the distance is set to value larger than zero then all direct neighbors are included. (3) Number of periods for which the optimization should be performed. (4) Length of a period in years. (5) Harvest flow (the differences in harvest volumes between periods). The optimization tries to maximize total harvest volumes over the periods. If the harvest flows would be larger than the one set, some of the harvest units are not assigned to any of the periods. The flow diagram is shown in Fig. 4.

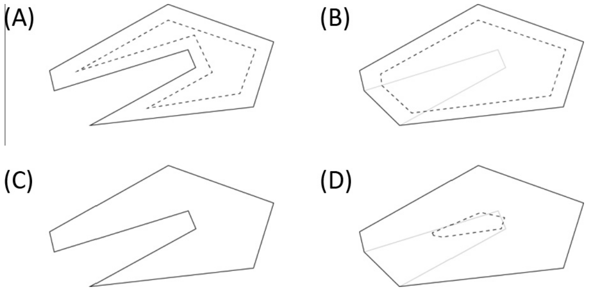

Fig. 3. An example of checking the maximal width of a polygon by constructing inner buffer. (A) A polygon with 25 m inner buffer, (B) convex hull of the same polygon with 25 m inner buffer, (C) polygon with 50 m inner buffer (in this case it was not possible to create such buffer), and (D) convex hull of a polygon with 50 m inner buffer.

After parameters for optimization are all set, by push of a button the adjacency matrix is created and passed automatically to solver, which performs optimization. The optimization package Gurobi® (Gurobi Optimization, Inc., 2014) is used as a solver for defined optimization model. It is directly linked to Optimal through Java API. The mathematical programming methods are commonly used for solving harvest scheduling optimization because of the computational efficiency (Pukkala, 2002). Special kind of mathematical programming – mixed binary programming has been used in Optimal. Each variable in the model represents single harvest unit. Using binary variables the results for each harvest unit indicate whether it should be harvested in a given period or not. The model has been described in detail in Kašpar et al. (2013).

3. Case study

The case study is presented on 46.5 ha of mature Spruce forest stands. It is based on real data, which is used with the agreement of the forest management area owner, but to comply with the rules for protection of personal data, it is not identified more specifically. The stocking volume ranges from 264 to 758 m^3/ha with the average 540 m^3/ha and standard deviation 61 m^3/ha. The area has been divided into 92 harvest units. Several scenarios of harvest flow percentages (i.e. the differences in harvest volumes between consecutive periods) were created starting with 2% and going up to 100% harvest flow. The other parameters were set to fixed values for all scenarios: Maximal area of harvest unit 1 ha, Minimal area of harvest unit 0 ha, Minimal width of harvest unit 25 m, Maximal width of harvest unit 50 m, Harvest units were considered as neighbors up to 25 m distance, Planning was optimized for 3 periods each 10 years long. We did not include any artificial gaps in this exercise.

The results of spatial and temporal optimization for one of the scenarios are presented in Fig. 5, to show an example of graphical output. All the scenarios, in terms of harvested volume per period, are shown in Fig. 6. The differences in total harvested volume per each scenario are shown in Fig. 7.

The results of the case study present various scenarios, which can be used by forest manager to make well informed decision. On one extreme the 2% harvest flow scenario results in approximately equal harvest volume compared between periods, but it is for the cost of lower total harvest volumes (over the all periods). On the other side, the 100% harvest flow scenario brings higher total harvest volumes, but the harvest volumes are not equally distributed over the three periods. That would have negative consequences on the forest enterprise economy. The operating costs would not be balanced over the periods causing problems with

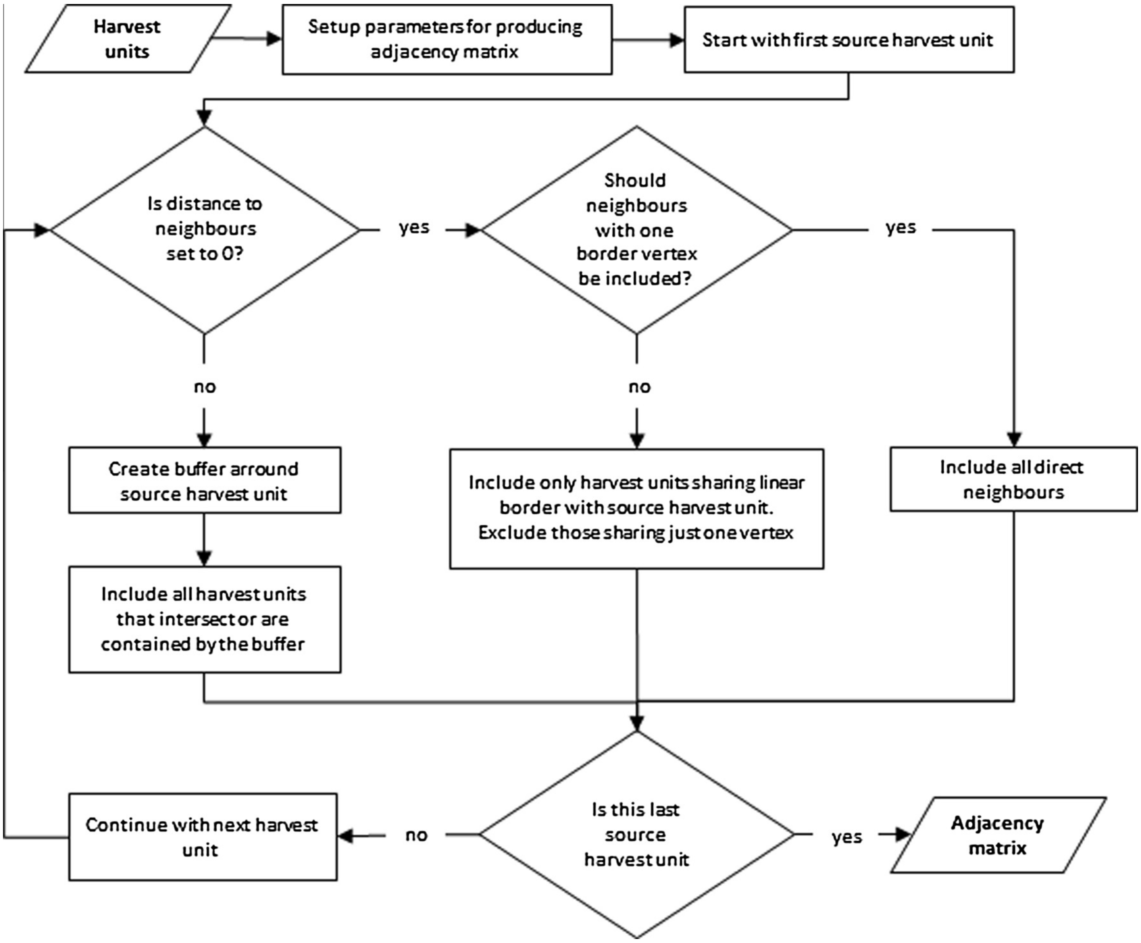

Fig. 4. Algorithm for automatic production of adjacency matrix from set of harvest units. A list of neighbors is created for each harvest unit and adjacency matrix is created from these lists. User can set parameters to tell the distance to which the polygons of harvest units are considered to be neighbors.

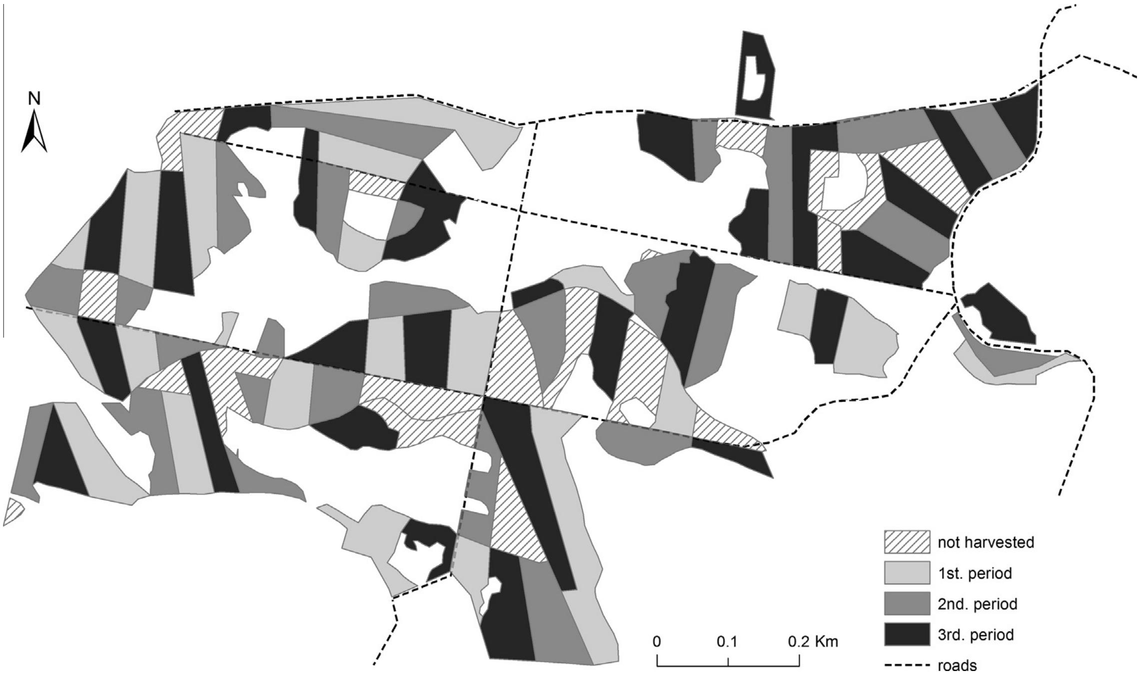

Fig. 5. The spatial and temporal distribution of harvests for 5% harvest flow.

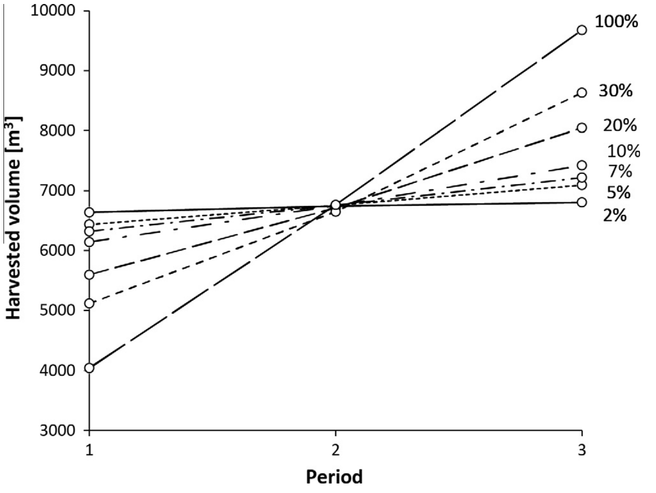

Fig. 6. Harvested volumes in planning periods according to harvest flow scenarios.

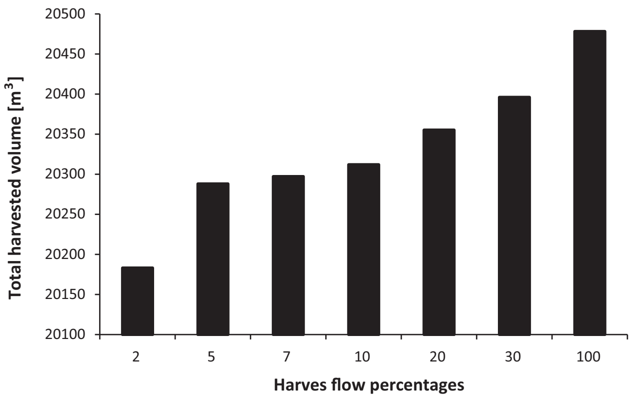

Fig. 7. Total harvest volumes for different harvest flows.

inefficient use of human resources and machinery. There would be also higher risk of forest damage due to over-aging of the forest stands, wind, pests, etc.

The aim of any forest manager should be to find the balance between high total harvest volumes and equality of the periodical harvests. In our case study we can see possible equilibrium around 5% harvest flow (see Fig. 7). The harvest flow is still low, ensuring equality of the volumes and the total harvest is already high enough to be comparable to higher harvest flow percentages.

4. Results and conclusions

Results of optimization are presented in the form of easy to understand map showing spatial distribution of harvest units in individual harvesting periods (see Fig. 5). User can repeat the simulation with different parameters and compare results. Our case study scenarios are built on various values of harvest flow. However, scenarios can be also built around different spatial and temporal constraints such as neighbor distances, number of periods and length of the periods.

We see the key value of Optimal software in bridging the gap between scientific understanding of harvest planning and real operational forest harvest planning. We involved forest managers in the design process and tried to create user interface as simple as possible to be understandable to anybody without prior knowledge of the embedded algorithms. The main benefits of using Optimal software are the speed in which the manager can create various scenarios, less guesswork and biased estimations involved in the decision process and compatibility with industry standard formats (shapefiles).

There are some limitations with current version, which we plan to overcome in next development. Optimal uses Java SDK for ArcGIS desktop extensions. By using the ArcGIS desktop functions for editing polygons we saved the development time, but at the same time we bound the application to proprietary software. In case the application should be used by forest managers who do not have ArcGIS license, it would presents additional cost that might limit the use of the application. Therefore our future plan is to move from desktop solution to server/client solution where the users will not be required to buy or even install anything on their computers. The users will then interact with the application through web browser. This will ensure easier and faster deployment of new versions, better monitoring of application use, but most importantly larger base of application users.

The optimization software Gurobi® is used as a MIP solver. We used academic version for the development and case study. In further development we plan to include not only Gurobi®, but optionally also other solvers.

In terms of internal algorithms the future development should go towards implementation of other forest management systems, such as e.g. shelter wood system.

To conclude, we built the Optimal to help forest manager to make well informed and efficient decisions faster than using traditional estimations.

Software availability

Name of software: Optimal. Extension is available on request to: vopenka@fld.czu.cz, kasparj@fld.czu.cz. Developers: Petr Vopěnka, Jan Kašpar. Contact address: Czech University of Life Sciences Prague, Kamycka 129, Praha 6 – Suchdol, Czech Republic. E-mail: vopenka@fld.czu.cz, kasparj@fld.czu.cz.

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).

References

Balzter, H., Braun, P.W., Köhler, W., 1998. Cellular automata models for vegetation dynamics. Ecol. Model. 107, 113–125.

Baskent, E.Z., Keles, S., 2005. Spatial forest planning: a review. Ecol. Model. 188, 145–173.

Baskent, E.Z., Kelles, S., 2006. Developing alternative wood harvesting strategies with linear programming in preparing forest management plans. Turk. J. Agric. For. 30, 67–79.

Bettinger, P., Boston, K., Siry, J.P., Grebner, D.L., 2009. Forest Management and Planning. Elsevier Inc.

Černý, M., Pařez, J., Malík, Z., 1996. Yields and Enumeration Tables for the Main Tree Species (Spuce, Pine, Beech, Oak). The Institute for Forest Ecosystem Research, Jílové u Prahy (in Czech).

Cho, S., Newman, D.H., 2005. Spatial analysis of rural land development. For. Policy Econom. 7, 732–744.

Gurobi Optimization, Inc., 2014. Gurobi Optimizer Reference Manual, Version 6.0. Available at: <http://www.gurobi.com/documentation/>.

Kašpar, J., Marušák, R., Vopěnka, P., 2013. Comparison of two alternative optimization techniques for spatial harvest planning. Sci. Agric. Bohemica 44, 90–96.

Kurttila, M., 2001. The spatial structure of forests in the optimization calculations of forest planning – a landscape ecological perspective. For. Ecol. Manage. 142, 129–142.

Lappi, J., Lempinen, R., 2013. J- Users' Guide. (In).

Murray, A.T., 1999. Spatial restrictions in harvest scheduling. For. Sci. 45, 45–52.

Ohman, K., Eriksson, L., 2002. Allowing for spatial consideration in long-term forest planning by linking linear programming with simulated annealing. For. Ecol. Manage. 161, 221–230.

Ohman, K., Lamas, T., 2003. Clustering of harvest activities in multi-objective long-term forest planning. For. Ecol. Manage. 176, 161–171.

Ohman, K., Lamas, T., 2005. Reducing forest fragmentation in long-term forest planning by using the shape index. For. Ecol. Manage. 212, 346–357.

Pukkala, T., 2002. Multi-Objective Forest Planning. Kluwer, Dordrecht.

Richards, E.W., Gunn, E.A., 2000. A model and tabu search method to optimize stand harvest and road construction schedules. For. Sci. 46, 188–203.

Sessions, J., Sessions, J., 1988. SNAP-A scheduling and network analysis program for tactical harvest planning. In: Proceeding of 1988 International Mountain Logging and Pacific Northwest Skyline Symposium, December. pp. 12–16.

Wikström, P., Edenius, L., Elfving, B., Eriksson, L.O., Lämås, T., Sonesson, J., Öhman, K., Wallerman, J., Waller, C., Klintebäck, F., 2011. The heureka forestry decision support system: an overview. Math. Comput. For. Nat. – Res. Sci. 3, 87–95.