10th International Conference of Modeling and Simulation - MOSIM'14 – November 5-7 - Nancy - France
"Toward circular Economy"

A MIXED-INITIATIVE SYSTEM FOR INTERACTIVE TACTICAL SUPPLY-CHAIN OPTIMIZATION

François Chéné, Jonathan Gaudreault, Claude-Guy Quimper

FORAC Research Consortium – Université Laval – Québec, Canada, G1V 0A6
Corresponding author: jonathan.gaudreault@forac.ulaval.ca

ABSTRACT: Tactical supply-chain planning consists of deciding the amount of products produced, consumed, stored or transported for each period of a given time frame with a precise objective to optimize. Many mathematical models have been proposed but most of the time they do not exploit decision makers intuition and preferences. Mixed-Initiative Systems (MIS) aim to solve problems in a different way, coordinating the interaction of intelligent automated agents and humans. MIS were developed for discrete combinatorial optimization but recently, a novel MIS approach has been proposed for problems showing a linear structure, like tactical supply-chain optimization. This approach allows a decision maker to explore a space of optimal solutions. In this paper, this scheme is improved in order to allow a user to explore near-optimal solutions as well. Furthermore, instead of interacting with lowly evocative bar charts, the decision maker uses the strongly visual map-based LogiLab user interface which allows him to view/modify in real time the flow between businesses of a forest-products supply chain.

KEYWORDS: Forest-products supply chain; Tactical planning; Mixed-Initiative Systems; Linear optimization

1 INTRODUCTION

Operation planning requires profound knowledge, solid experience, acute instinct, and sufficient information. While computers excel in treating large amounts of information, they often lack the knowledge and experience of human managers (Klau et al., 2010). This has led some researchers to propose *Mixed-Initiative Systems* (MIS) that promote the interaction of a human decision maker with an automated computing agent. (Allen, 1999; Hearst, 1999)

Classical Mixed-Initiative Systems were developed for discrete combinatorial optimization. Recently, Hamel et al. (2012) proposed a MIS approach (Hamel's method) for linear problems like supply chain tactical optimization. Within Hamel's approach, the system pre-computes a set of optimal solutions and the user can explore the solution space by interacting with bar charts showing the current solution. Following each variable modification requested by the user, the system re-computes and displays in real time a new optimal solution.

In this paper, this scheme is improved in order to allow the user to explore near-optimal solutions as well. Furthermore, instead of interacting with bar charts, the decision maker uses the map-based LogiLab user interface which allows him to view and modify in real time the flow between businesses of a forest-products supply chain. The user also has the option of editing solutions using aggregated results (e.g. annual flows), or detailed results (e.g. flows per period).

The remainder of the paper is organized as follows. Section 2 presents preliminary notions on supply chain optimization, Mixed-Initiative Systems (MIS), and Hamel's method for interactive flow optimization. Section 3 presents the LogiLab forest-products tactical supply-chain optimization system. We propose some modifications to its graphical user interface that allow using it as an MIS exploiting Hamel's method. Then, in Section 4, we expand Hamel's methods in order to allow the user to explore an expended set of (near-optimal) solutions. Section 6 presents a proof of concepts together with experimental results for a forest-products supply chain. Section 7 concludes the paper.

2 PRELIMINARY NOTIONS

2.1 Supply Chain Optimization (SCO)

Supply chains are formed by a set of business units where processes consume resources to generate products. Resources can be local to the business units, like the hours that can be worked, or products of other processes, which can occur in other business units. They also include external resource suppliers and clients. Generally, products must be transported between business units.

Tactical supply-chain planning consists of computing the amount of products produced, consumed, stored or transported for each period of a given time frame with a precise objective to optimize. Common objectives are costs minimization or profit maximization.

Such problems can easily be modeled using a linear programming model. They can be solved using well-known algorithms, like the Dantzig (1955) simplex algorithm.

Tactical supply-chain planning is a very important concern in the forest-products industry. This industry is facing divergent product flows (Haartveit et al., 2004). A single business unit produces many products at the same time from a single piece of raw material. This leads to an important interdependency between business units (e.g. forest operations supply many different types of industries at the same time, a sawmill supplies lumber to remanufacturing plants as well as chips to paper mills, etc.). Many mathematical models have been proposed (Jerbi et al., 2012; Singer and Donoso, 2007) but most of the time they do not exploit decision makers' intuition and preferences.

2.2 Mixed-Initiative Systems (MIS)

In order to complete a complex task, one may gather multiple specialists of different trades relevant to the problem at hand. It is quite difficult to model preferences of many decision makers. They are often unaware of their own preferences before seeing a solution that violates them. Mixed-initiative systems aim to solve problems in a different way, coordinating the interaction of intelligent automated agents and humans (Hearst, 1999).

Involving humans in the search for an optimal solution provides multiple benefits. Humans generally outperform computers in visual perception and strategic thinking. They can also justify and improve solutions in which they participate, and they can implement their preferences and knowledge of the real world without complex and sometimes near-impossible mathematical modeling. Finally, human participation generates a stronger trust in the produced solution (Klau et al., 2010).

Most MIS-related research targets discrete combinatorial optimization problems such as timetabling (Kun and Havens, 2005), space mission planning and scheduling (Ai-Chang et al., 2004; Bresina and Morris, 2006, 2007), air traffic control (Guiost et al., 2004), military applications (Lenor et al., 2000; Linegang et al., 2003), etc.

2.3 MIS for linear optimization

Hamel et al. (2012) proposed an MIS approach for linear optimization. The system allows the user to visualize an implicit sub-space of optimal solutions for a given set of variables. The user can interactively increase or decrease the value of a variable and the system reacts by computing and displaying, in real time, the new sub-space of optimal solutions (Figure 1).

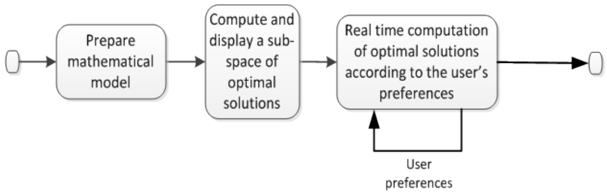

Figure 1. MIS concept applied to linear programming (Adapted from Hamel et al. 2012)

The system works as follows. It uses a classical linear programming solver to compute an optimal solution to a problem *P*. Once the optimal objective value is known, it searches for other optimal solutions by creating a new instance of the problem, called *P'*, by adding a constraint forcing the objective value to be equal to the optimal value found for *P*. The space of *feasible* solutions of *P'* is therefore equivalent to the space of *optimal* solutions of *P*. For each of the *n* variables x_i displayed to the user, the system searches for a solution $\underline{S_i}$ that minimizes x_i and a solution $\overline{S_i}$ that maximizes x_i while preserving the optimality of the solutions. These $2n$ computations can be performed in parallel using a supercomputer.

Then, one can display on the chart (see

Figure 2) the *range of optimality* for each variable x_i (the minimum and maximum values the variable can take in an optimal solution).

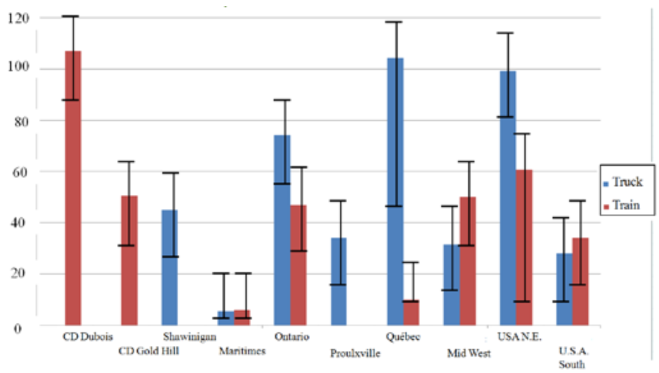

Figure 2. An optimal solution showing the range of optimality for the main decision variables

A chart like the one depicted in

Figure 2 allows the user to see the flexibility (range of optimality) he has for any of the variables of interest. He can thus change the value of a variable within the optimality zone of that variable, for example by dragging up or down the top of a bar in the chart. Upon the modification of a variable, the system adjusts the value taken by the other variables such that the solution remains feasible and optimal. Normally, this operation would require a lengthy complete re-optimization of the problem. However, we know that it is possible to compute a new optimal solution by generating a convex combination of the optimal solutions obtained from the previous section. To find a solution with a given variable assigned to a specific value, it is sufficient to compute a new convex combination of extreme solutions.

This can be done in real time and it allows instantly refreshing the chart and displaying the impact of the user modification on the other variables. The user can thus *navigate* through the solution space, successively modifying several variables until a suitable solution is found.

Figure 3 illustrates this idea for a small problem with two variables. The gray polygon represents the set of convex combinations formed by optimal solutions minimizing and optimizing each of the variables[1].

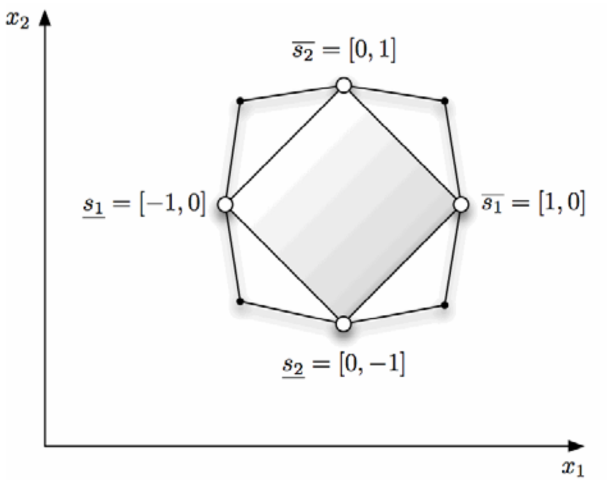

Figure 3. Available optimal zone using convex combinations

When the user modifies a variable, the system not only looks at any point in the sub-space such that the modified variable takes the desired value, but it also looks for a solution such that the other variables move as little as possible so that the overall system appears to be *stable*. Given a solution *x*, a variable x_i, and a value *v*, finding a new solution x' where $x'_i = v \neq x_i$ while changing as little as possible the values of the other variables is an optimization problem.

Since we aim to have the human agent to interact in real-time with the system, responsiveness was an important criterion, Hamel's method looks for a solution in the convex hull of the 2*n* precomputed solutions $\underline{s_1}, \overline{s_1}, \ldots, \underline{s_n}, \overline{s_n}$, a much smaller problem than the original one. Hamel proposed what he called the *triangular heuristic*. First, it determines if the user asked for the variable x_i to be increased ($v > x_i$) or decreased ($v < x_i$). If the variable is increased, he computes the unique convex combination between the current solution *x* and $\overline{s_i}$ that intersects the hyperplane $x_i = v$. If the variable is decreased, he computes the unique convex combination between the current solution *x* and $\underline{s_i}$ that intersects the plane $x_i = v$. In other words, if the decision maker increases x_i, he sets

$$x' = \alpha x + (1 - \alpha)\overline{s_i}$$

where

$$\alpha = \frac{\overline{s_{ii}} - v}{\overline{s_{ii}} - x_i.}$$

If the decider decreases the variable x_i, he sets

$$x' = \alpha x + (1 - \alpha)\underline{s_i}$$

where

$$\alpha = \frac{\underline{s_{ii}} - v}{s_{ii} - x_i.}$$

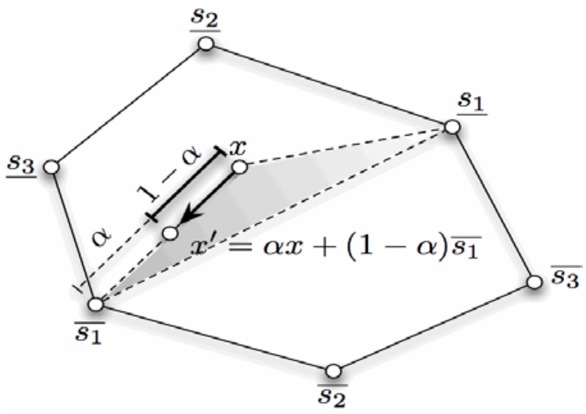

Figure 4. An example of the triangular method

In the next sections, we explain how Hamel's method can be adapted to allow the user to explore an expanded set of (near-optimal) solutions. We also show how we adapted the user interface in order to allow the user to interact with flows on a map instead of bar charts.

3 PROPOSED GRAPHICAL USER INTERFACE (GUI) FOR INTERACTIVE SUPPLY CHAIN TACTICAL OPTIMIZATION

The FORAC Research Consortium developed a software named LogiLab to optimize and display the design and planning of forest-products supply chains. The

[1] This example (Figure 3) also shows that there could be optimal solutions (in white) not covered by this sub-space (in gray). This is why we say that the user navigates in a space (formed by convex combination of our extreme solutions) that is in reality a sub-space of the optimal solutions.

mathematical optimization model used by LogiLab is described by Jerbi et al. (2012).

Since LogiLab's data presentation is strongly evocative and natural for planners (See figure 5), we decided to use it as a basis for our interactive system. In the original LogiLab, the products flows from one business unit to another are presented by arcs connecting the corresponding nodes. The thickness of each arc is proportional to the volume of products that flows from its origin to its destination.

As we now want to display the "range of optimality" of each individual variable, we proposed replacing each arc by a "pipe". According to this metaphor, the diameter of the pipe represents the maximum value the flow variable can take. The colored part of the pipe represents the current level of goods that flows through the pipe (i.e. value of the variable in the current optimal solution). To represent the minimal value that allows for the solution to be optimal, the bottom part of the flow in the pipe is displayed with a darker color (see Figure 6).

The simplest and most intuitive way to allow the user to interact with the flows (in order for him to adjust the value of a variable) is simply to allow him to click on the pipe and then drag the mouse within the range of optimality. However, most flows tend to be too small to enable proper manipulation. Therefore, when the user rolls the cursor over a flow, an enlarged cross-section appears and allows easier manipulations (see Figure 4).

In the original LogiLab, the percentage of the total capacity of a business unit used is displayed using a rectangle similar to a progression bar (look at the colored rectangle under each icon in Figure 5).

We added visualization of the range of optimality data for these variables using a color code similar to what we proposed for the flow variables.

Figure 6 presents the graphical user interface integrating those concepts. It presents an interaction scenario: (1) The user rolls over a flow; the zoom bubble (that resembles the cross-section of a pipe) appears and the user is allowed (2) to set a new value for this flow variable. Finally (3), the system updates in real time the value of the other variables, thus computing a feasible solution that is still optimal.

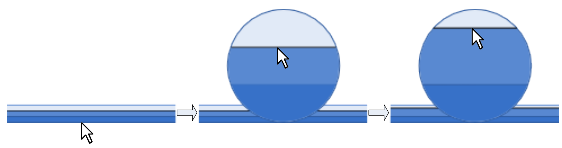

Figure 4. Interaction with flows within the proposed system

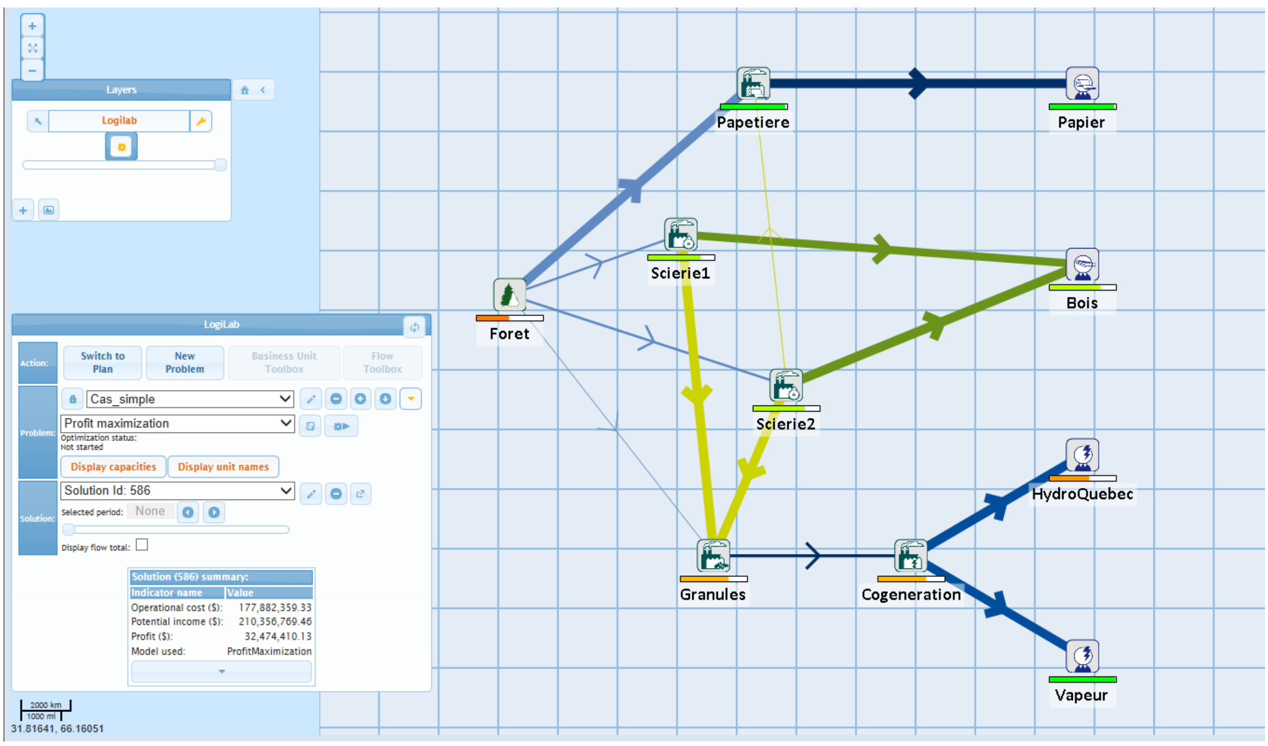

Figure 5. LogiLab graphical user interface

(1)
(2)
(3)

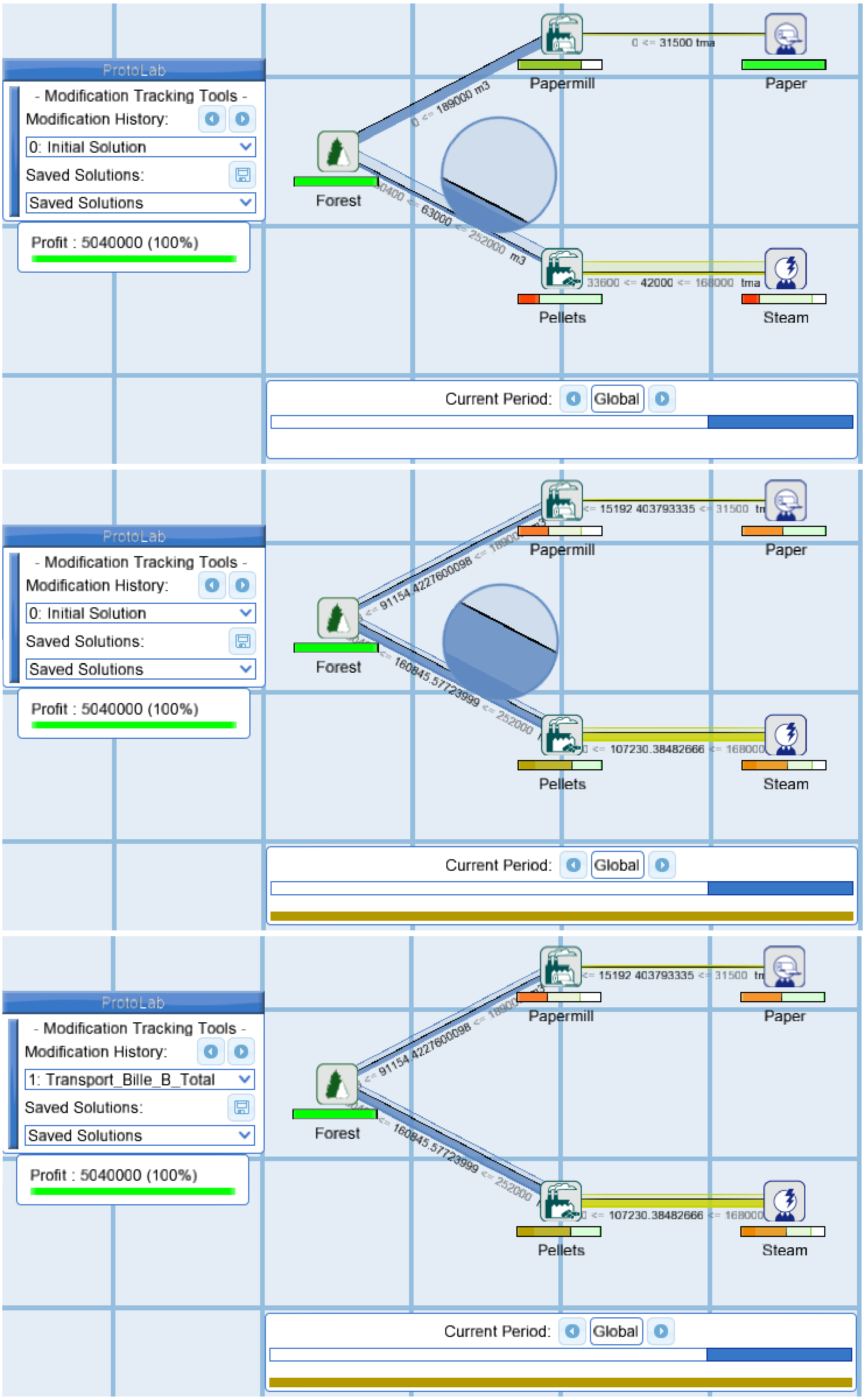

Figure 6. Proposed graphical user interface for interactive supply chain tactical optimization

4 INTRODUCING OPTIMALITY TOLERANCE (GAP) INTO HAMEL'S METHOD

Hamel's original method, even coupled to our new graphical user interface, only allows exploring optimal solutions. However, decision makers may be interested in exploring near-optimal solutions too (as the data used to model the problem are often rough estimates, it would be pointless to exclude solutions that decrease profit by only a few dollars).

The first (naive) approach we tried to modify Hamel's approach was the following. When computing solutions minimizing/maximizing a given variable, we tolerate a gap between the new solution's profit and the optimal solution of the original problem.

However, this approach does not work very well when using the triangular heuristic. Interaction with the system revealed frequent cases where adjusting a variable to values within the optimal space would return a near-optimal solution while some existing optimal solutions would clearly be a better choice. We aim to explain the cause of this behavior and to propose another approach.

With the triangular heuristic, as the values of the variables are interpolated between two solutions, so are the profit values associated to these solutions. Therefore, if one solution is optimal but not the other, unless the weight of the optimal solution is 0 and the weight of the non-optimal solution is 1, then the solution resulting from the combination of these solutions will not be optimal.

This was not an issue with Hamel's original method where all solutions were optimal. However, with the modified approach, we combine non-optimal solutions. This result is that the system (1) returns a non-optimal solution as soon as a variable is modified towards an extreme value associated with a non-optimal solution even if the target value is within the optimal range for that variable (see Figure 7) and (2) once a non-optimal solution is reached, the system does not return an optimal solution unless a variable is set to an extreme value associated with an optimal solution.

This behavior is explained in Figure 7. In this example, the green area is the real optimal solution space. The yellow area shows how this space expands when we tolerate an optimality gap. In this example, we suppose the system first displays the solution for which y is maximized and then the user asks to gradually decrease x. As soon as x starts to decrease, the triangular heuristic gets a sub-optimal solution as it interpolates between $\overline{y}$ and $\underline{\underline{x}}$. This is a pity as it is clear that for some values for x we could still have an optimal (green) solution.

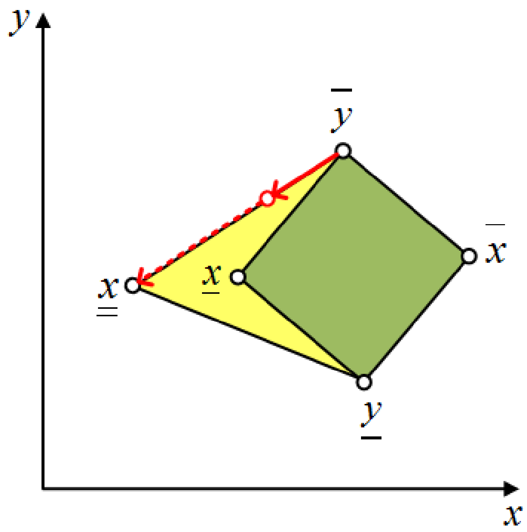

Figure 7. Behavior of Hamel's original Triangular heuristic when provided with a solution space with optimality gap

The situation is even worse for the following reason. Suppose we use a solver (e.g. the simplex method) to find a solution minimizing/maximizing a given variable, with the constraint that the profit of the network should be at least 95% of the profit of the original optimal solution. The simplex could return a solution minimizing/maximizing the variable that has a profit of 95% even if there exists another solution where the variable takes the exact same value while showing a profit of 100%.

To palliate this situation, we propose a multi-stage approach in the next subsection.

4.1.1 Proposed multi-stage approach

We want the system to return an optimal solution whenever possible. To achieve this, we need two pairs of solutions for each variable x_i the user is interested to modify. As in Hamel's original method, one pair of solutions $\{\underline{x_i}, \overline{x_i}\}$ minimizing/maximizing the variable value while preserving optimality and another pair $\{\underline{\underline{x_i}}, \overline{\overline{x_i}}\}$ minimizing/maximizing the variable value obtained when the optimality constraint is relaxed (according to some gap specified by the user).

We also modified the triangular heuristic to interpolate using tolerated extreme points $\underline{\underline{x_i}}$ and $\overline{\overline{x_i}}$ only when necessary.

This behavior is illustrated in Figure 8. In this example, we suppose the system first displays the solution for which y is maximized. Then the user asks to gradually decrease x. As long as x is greater than the value it has in solution $\underline{x}$, we interpolate between the solutions $\overline{y}$ and $\underline{x}$ and the solution is still optimal. Past that point, we interpolate between the solutions $\underline{x}$ and $\underline{\underline{x}}$.

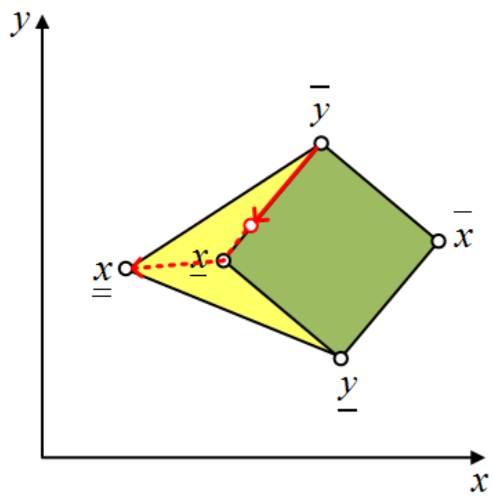

Figure 8. Behavior of the multi-stage approach

5 PROOF OF CONCEPT

In this section, the proposed multi-stage approach is compared to the original Hamel method and to the Hamel method with tolerance.

We consider a small linear optimization problem (wood flow supply chain) that we model with LogiLab. This problem is graphicly represented in Figure 6. Logs are harvested from a forest and can either be transformed into paper or pellets. Both products cost the same to produce and generate the same income per log. Paper demand is below production capacity, implying that a minimal amount of pellets must be produced to maximize the profit. We suppose a planning horizon of 3 periods and we have a total of 37 variables.

Using the *Hamel's original approach*, the range of optimality of the variables is rather small.

The decision maker now wants to interactively explore/discover near optimal solutions (max gap. 5 %) that best meet his preferences. Introducing tolerance should allow decreasing pellet production, and/or reduce forest harvesting volumes.

Using the *Hamel with gap tolerance* approach, variables should provide a wider range of optimality (thanks to the allowed 5 % gap) but most manipulation of a variable by the user should lead to non-optimal solutions (even though variables are manipulated within their range of optimality).

With the *multi-stage approach*, we should have the same range of optimality for the variables, but a small modification to these variables should lead to optimal solutions.

In the following experiment, we formally compare the approaches according to these criteria: (1) range of optimality of the variables, and (2) the optimality of the solutions that are computed after a modification to a variable.

For each method, we computed the range of optimality of each variable. Then, for each variable, we incremented/decremented the variable and measured the gap between the new solution and the original one.

Figure 9 shows how the range of optimality of individual variables are extended (in comparison to the original Hamel method) when we allow a 5 % gap (results are the same for Hamel with gap tolerance and the multi-state approaches). We see no increase for 16 variables, a 5% increase for 6 variables (including of course the objective-function "profit" variable) and a 15% increase for 15 other variables.

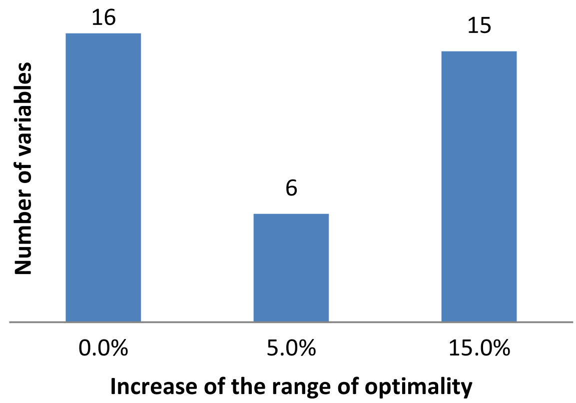

Figure 9. Increase of the range of optimality of the individual variables when provided with a 5% gap on the solution objective function

The ill effects of the *Hamel with tolerance gap* approach appeared as expected. Out of the 72 "extreme" solutions, ($\underline{\underline{x}}$ and $\overline{\overline{x}}$ minimizing/maximizing the 36 manipulable variables), 52 were non-optimal.

Except for the pellet production and the forest harvesting variables, even small modification to a variable leads to a sub-optimal solution (in which pellet production and forest harvesting decrease). Consequently, the system gives the users the false impression they cannot modify these variables without altering optimality.

Using the *multi-stage* approach, out of the 144 solutions corrresponding to $\underline{x}, \overline{x}, \underline{\underline{x}}, \overline{\overline{x}}$, only 20 were sub-optimal (they correspond to the situations where $\underline{\underline{x}} < \underline{x}$ or $\overline{\overline{x}} > \overline{x}$). All other manipulations of a variable in the range between $\underline{x}$ and $\overline{x}$ lead to optimal solution. This can be verified for any variable but we report results for one specific variable in Figure 10. For the purpose of our demonstration, we selected a variable that has some range of optimality even when the optimality gap is zero.

The blue curve shows that using the original Hamel method, we can modify the variable from $\overline{x}$ to $\underline{x}$ and from $\underline{x}$ to $\overline{x}$ and the new computed solution is always optimal.

Using the Hamel method with gap tolerance (in red) we leave the optimality region as soon as we move from $\underline{x}$ toward $\underline{\underline{x}}$.

With the multi-stage approach (in green), the solution is optimal between $\overline{x}$ and $\underline{x}$. We leave optimality only if we go below $\underline{x}$ toward $\underline{\underline{x}}$, which is the expected behavior. A similar behavior is shown when we move back to $\overline{x}$.

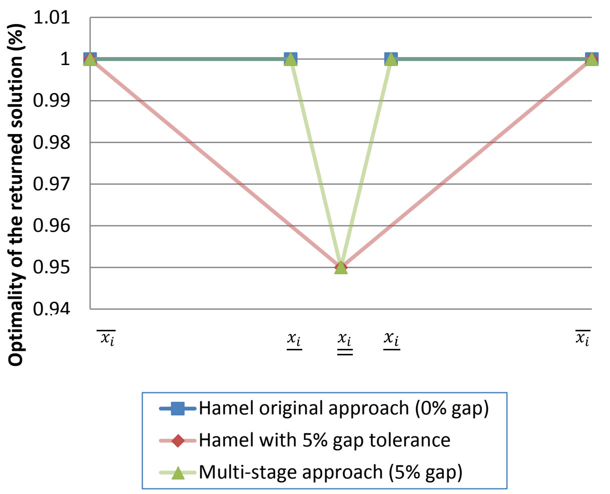

Figure 10. Comparison of optimality of returned solutions during interaction. For this variable, the solutions for $\overline{x}$ and $\overline{\overline{x}}$ were identical, this is why $\overline{\overline{x}}$ does not appear in the chart.

6 CONCLUSION

Interactive linear optimization with human implication in a Mixed-Initiative System offers promising possibilities. Previous research has produced interesting results for the linear optimization problem. We deemed it appropriate to pursue this research through improvement of the mechanism by introducing optimality tolerance.

We proposed a new user interface for interactive supply chain optimization based on the LogiLab graphical user interface, a software developed for supply chain optimization in the forest-products industry.

We presented an experimentation that illustrates how the user experience of the decision makers and the solution space he can explore in real time can be expanded using the proposed approaches.

REFERENCES

Ai-Chang, M., Bresina, J., Charest, L., Chase, A., Hsu, J.C.J., Jonsson, A., Kanefsky, B., Morris, P., Kanna, R., Yglesias, J., Chafin, B.G., Dias, W.C., Maldague, P.F., 2004. MAPGEN: mixed-initiative planning and scheduling for the Mars Exploration Rover mission. IEEE Intelligent Systems 19, 8-12.

Allen, J.E., 1999. Mixed-initiative interaction. IEEE Intelligent Systems 14, 14-16.

Bresina, J.L., Morris, P.H., 2006. Mission operations planning: beyond MAPGEN, Second IEEE International Conference on Space Mission Challenges for Information Technology. IEEE Comput. Soc, Pasadena, California, pp. 477-484.

Bresina, J.L., Morris, P.H., 2007. Mixed-initiative planning in space mission operations. AI Magazine 28, 75-88.

Dantzig, G.B., 1955. Linear programming under uncertainty. Management science, 197-206.

Guiost, B., Debernard, S., Poulain, T., Millot, P., 2004. Supporting air-traffic controllers by delegating tasks, Proceedings of the 2004 IEEE International Conference on Systems, Man and Cybernetics. IEEE, The Hague , Netherlands, pp. 164-169.

Haartveit, E.Y., Kozak, R.A., Maness, T.C., 2004. Supply chain management mapping for the forest products industry: Three cases from western Canada. Journal of Forest Products Business Research 1, 31.

Hamel, S., Gaudreault, J., Quimper, C.-G., Bouchard, M., Marier, P., 2012. Human-machine interaction for real-time linear optimization, 2012 IEEE International Conference on Systems, Man, and Cybernetics (IEEE SMC 2012), Seoul, Korea, pp. 673-680.

Hearst, M.A., 1999. Mixed-Initiative Interaction. IEEE Intelligent Systems 14, 14.

Jerbi, W., Gaudreault, J., D'Amours, S., Nourelfath, M., Lemieux, S., Marier, P., Bouchard, M., 2012. Optimization/simulation-based framework for the evaluation of supply chain management policies in the forest product industry, IEEE International Conference on Systems, Man, and Cybernetics (IEEE SMC 2012), Seoul, Korea, pp. 1742-1748.

Klau, G., Lesh, N., Marks, J., Mitzenmacher, M., 2010. Human-guided search. Journal of Heuristics 16, 289-310.

Kun, W., Havens, W.S., 2005. Modelling an academic curriculum plan as a mixed-initiative constraint satisfaction problem, Proceedings of the 18th Conference of the Canadian Society for Computational Studies of Intelligence (LNCS #3501). Springer-Verlag, Victoria, Canada, pp. 79-90.

Lenor, T., Lewis, M., Hahn, S., Payne, T., Sycarn, K., 2000. Task characteristics and intelligent aiding, Proceedings of the 2000 IEEE International Conference on Systems, Man and Cybernetics. IEEE, Piscataway, NJ, USA, pp. 1123-1127.

Linegang, M., Haimson, C., MacMillan, J., Freeman, J., 2003. Human control in Mixed-Initiative Systems: lessons from the MICA-SHARC program, Proceedings of the 2003 IEEE International Conference on Systems, Man and Cybernetics. IEEE, Washington, D.C., pp. 436-441.

Singer, M., Donoso, P., 2007. Internal supply chain management in the Chilean sawmill industry. International Journal of Operations & Production Management 27, 524-541.