Proceedings of the Ninth International
Symposium on Combinatorial Search (SoCS 2016)

An Improved Metaheuristic Algorithm for Maximizing Demand Satisfaction in the Population Harvest Cutting Stock Problem

Laura Climent, Barry O'Sullivan, and **Richard J. Wallace**
Insight Centre for Data Analytics, Department of Computer Science, University College Cork, Ireland
{laura.climent, barry.osullivan, richard.wallace}@insight-centre.org

Abstract

We present a greedy version of an existing metaheuristic algorithm for a special version of the *Cutting Stock Problem* (CSP). For this version, it is only possible to have indirect control over the patterns via a vector of continuous values which we refer to as a *weights vector*. Our algorithm iteratively generates new weights vectors by making local changes over the best weights vector computed so far. This allows us to achieve better solutions much faster than is possible with the original metaheuristic.

Introduction

In this paper, we deal with a variant of the *Cutting Stock Problem* CSP that arises in the forestry industry; this problem is a type of population harvesting, such as harvesting of plants or lumber. Viewed from an optimization perspective, forest harvesting is a bilevel optimization problem. The first lower level is the individual tree stem which must be cut, or "'bucked" using forestry terminology, to optimize the total value of the log products produced; this is referred to as "bucking-to-value". The second level is the stand or forest, which we consider as a unit. The objective is to minimize the difference between the *global products pattern*, i.e. the total amounts of products cut from the stand or forest, and the original customer demand; this is referred to as "bucking-to-demand".

Bucking-to-value is a recursive problem, i.e. maximize value by cutting the first product and then maximize the value of the remainder. Therefore it can be solved by dynamic programming (DP) (Pnevmaticos and Mann 1972). The best current approach for the bucking-to-demand is to simulate the cutting of a *sample* of tree stems, starting with the original price (or weight) vector and adjusting it to better meet overall customer demand. After carrying out a simulation the chosen weights vector is used by harvesting machines to cut the real raw material elements, i.e. the full population of tree stems. In this paper we extend a well-known adaptive control heuristic (Murphy, Marshall, and Bolding 2004) by incorporating new greedy features.

Population Harvesting CSP

In this problem we have a fixed number $r = 1 \dots |R|$ of raw material pieces each with its own dimensions $\sigma_r \in S$ (e.g. tree stems of specific dimensions). Due to the different dimensions of the raw material pieces, the patterns differ for each piece. During search the metaheuristic produces a *global* pattern $p = \langle a_1, \dots, a_{|\mathcal{M}|}\rangle$, where $a_j \in [0, 1]$ represents the percentage of units of product $m_j \in \mathcal{M}$ cut from the set of raw material pieces R. The weight vector is $v \in \mathbb{R}_+{}^{|\mathcal{M}|}$, where each v_j represents the value associated with the product $m_j \in \mathcal{M}$. The $\mathcal{A}$ algorithm, usually implemented as a DP procedure, simulates the cutting of each raw material piece so as to maximize its total value based on the values of the products (v). The mix of products that we obtain represents a global product pattern. $\mathcal{A}$ can be represented by the following mapping function:

$$\mathcal{A}(\mathcal{M}, < \sigma_1, \dots, \sigma_{|R|} >, v) \rightarrow p. \tag{1}$$

The demands for products $d \in [0, 1]^{|\mathcal{M}|}$ is a vector where d_j represents the percentage of units of product $m_j \in \mathcal{M}$ that are demanded. To measure the similarity of demanded and obtained amounts we use the *Apportionment Degree* (AD) (Kivinen, Uusitalo, and Nummi 2005). The AD function maps the difference between the global products pattern p and the demand d to $[0, 100]$, where 100 means a perfect match. However, typically there is waste from the cutting process, therefore the optimal pattern cannot reach the AD of 100. The AD is defined as follows:

$$AD(p, d) = 100(1 - \frac{\sum_{j=1}^{|\mathcal{M}|} |a_j - d_j|}{2}). \tag{2}$$

An Improved Adaptive Control Heuristic

In this paper we present enhancements to a well-known simulated annealing-like algorithm (SALA) from the literature (Murphy, Marshall, and Bolding 2004). SALA algorithms are characterized by an accept-worse-solutions criterion that allows a wider search in the solution space. Figure 1 is a step-by-step illustration of such a metaheuristic.

The new greedy features are related to the shaded modules of the diagram. The first shaded module is where a subset of

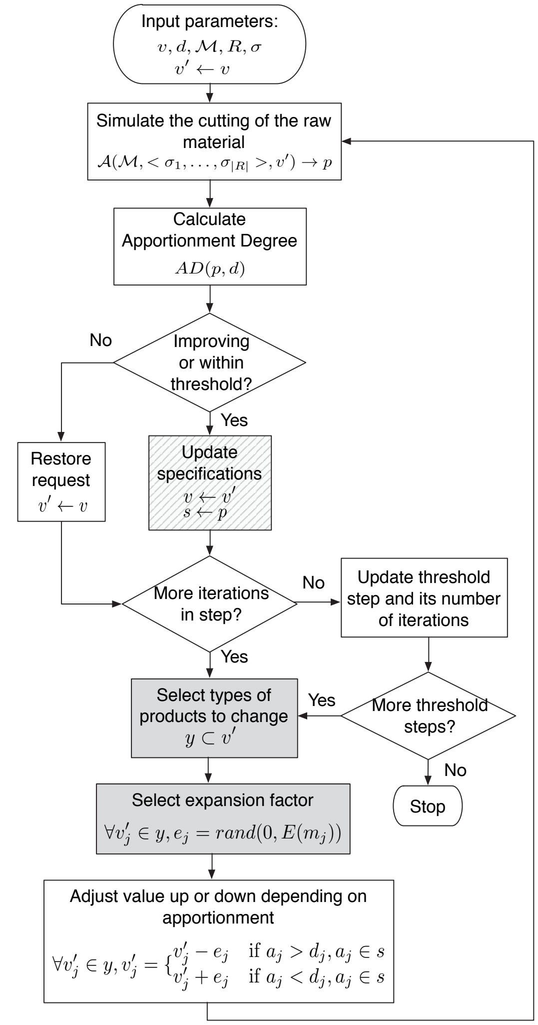

Figure 1: Metaheuristic for the Population Harvesting CSP.

variables y is selected for changing the current weight vector v'. In the original metaheuristic the selection is done randomly. Instead, we include a greedy feature that computes specific probabilities of selection for each variable, depending on the difference between the percentages of product obtained (a_j) and product demanded (d_j):

$$p(v_j) = \frac{|a_j - d_j|}{\sum_{i=1}^{|\mathcal{M}|} |a_i - d_i|}. \tag{3}$$

In the second shaded module, for each variable in y an expansion factor is selected randomly within the interval $(0, E(m_j)]$. In the original metaheuristic the same maximum expansion factor ($E(m_j)$) is used for all the product types. In contrast, we calculate a specific maximum expansion factor for each selected product based on its demand matching (magnified/reduced by $k \in \mathbb{R}_+$):

$$E(m_j) = k|a_j - d_j|. \tag{4}$$

Evaluation and Conclusions

In this section we compare the original metaheuristic with our extension using real forestry data from an industry partner. In Figure 2 we analyze a very high demand for the second cheapest product (85%) and equally low demand of each of the other three products (5% each). We used the same settings for the two metaheuristics: threshold steps in $\{0.050, 0.020, 0.010, 0.005, 0.001\}$, number of iterations for each threshold step in $\{5, 5, 5, 5, \infty\}$, and time-outs of $\{50, 100, 150, 200\}$ seconds. In addition, the initial vector of values v is the same for both techniques and only one type of product is selected as a subset y in each iteration. Specifically, for the original metaheuristic $E(m_i) = 2, \forall i \in \mathcal{M}$, and for the greedy metaheuristic $k = 5$ in Equation 4.

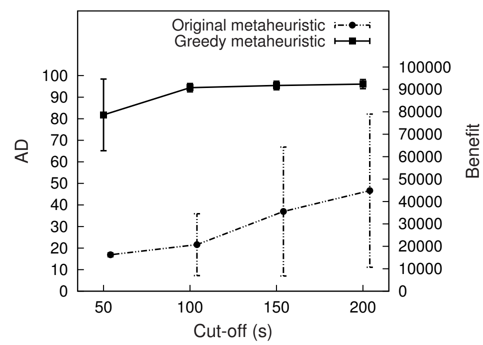

Figure 2: AD and benefit (€) for several times-outs.

Figure 2 shows the quality of the best global product patterns obtained. The improvement in performance of our greedy features is especially noticeable when the available computation time is limited, as in the case in many real-life on-line applications. In some cases the improvements led to benefits in the thousands of euros for the real-life forestry harvesting instance analyzed.

Future work will focus on applying this approach to other types of real-life CSPs.

Acknowledgments

This work is supported by Science Foundation Ireland (SFI) under grant SFI/12/RC/2289.

References

Kivinen, V.-P.; Uusitalo, J.; and Nummi, T. 2005. Comparison of four measures designed for assessing the fit between the demand and output distributions of logs. *Canadian journal of forest research* 35(3):693–702.

Murphy, G.; Marshall, H.; and Bolding, M. C. 2004. Adaptive control of bucking on harvesters to meet order book constraints. *Forest Products Journal and Index* 54(12):114–121.

Pnevmaticos, S. M., and Mann, S. H. 1972. Dynamic programming in tree bucking. *Forest Products Journal* 22:26–30.