Weighted Set Covering Problem for Cutting Stock Problems with Setup Costs

Benedikt Klocker

Algorithms and Complexity Group
Institute of Computer Graphics and Algorithms
TU Wien

PHD Seminar, 05 December 2016

Introduction

- Part of the Lodestar project
  - Finding optimal cutting patterns in 2d
  - Only allows guillotine cuts and has many more optional machine restrictions
- Very general solution approach for cutting stock problems of arbitrary dimensions

General Cutting Stock Problem

$E = \{1, \ldots, n\}$ …set of elements, each having a given size

$(d_i)_{i=1}^n \in \mathbb{N}^n$ …demand vector

T …set of sheet types, each having a given size, costs $c(t)$ and maximal amount $a(t)$

A *pattern* p is a structured collection of elements from E together with a sheet type $t_p \in T$. Each pattern p is associated with an element vector $(e_i^p)_{i=1}^n \in \mathbb{N}^n$, production costs c_p^{P} and setup costs c_p^{S}. The problem is now to find a set of feasible patterns P and amounts $a = (a_p)_{p \in P} \in \mathbb{N}^{|P|}$ which satisfies

$$\sum_{p \in P} a_p \cdot e_i^p \geq (=) d_i \quad \forall i \in \{1, \ldots, n\}$$

and for each sheet type t satisfies $\sum_{p \in P : t_p = t} a_p \leq a(t)$ and minimizes the costs

$$\sum_{p \in P} a_p \cdot c_p^{\mathrm{P}} + \left\lceil \frac{a_p}{s^{\max}} \right\rceil \cdot c_p^{\mathrm{S}}$$

Weighted Set Covering Problem

- Split Problem into two parts
- Part 1: Generate a large set of good candidate patterns P
- Part 2: Find the optimal amounts vector $a = (a_p)_{p \in P} \in \mathbb{N}^{|P|}$

Weighted Set Covering Problem for the Cutting Stock Problem

Let the set P be given. Find a vector $a = (a_p)_{p \in P} \in \mathbb{N}^{|P|}$ satisfying

$$\sum_{p \in P} a_p \cdot e_i^p \geq (=) d_i \qquad \forall i \in \{1, \ldots, n\},$$

$$\sum_{p \in P : t_p = t} a_p \leq a(t) \qquad \forall t \in T$$

which minimizes

$$\sum_{p \in P} a_p \cdot c_p^{\mathrm{P}} + \left\lceil \frac{a_p}{s^{\max}} \right\rceil \cdot c_p^{\mathrm{S}}.$$

MIP Formulation

$$\min_{(a_p)_{p\in P},(s_p)_{p\in P}} \sum_{p\in P} a_p \cdot c_p^{\mathrm{P}} + s_p \cdot c_p^{\mathrm{S}} \tag{1}$$

$$\text{s.t.} \sum_{p\in P} a_p \cdot e_i^p \geq (=) d_i \qquad \forall i \in \{1,\dots,n\} \tag{2}$$

$$\sum_{p\in P: t_p = t} a_p \leq a(t) \qquad \forall t \in T \tag{3}$$

$$s_p \cdot s^{\max} \geq a_p \qquad \forall p \in P \tag{4}$$

$$a_p \in \mathbb{N}, s_p \in \mathbb{N} \qquad \forall p \in P \tag{5}$$

Greedy Approach

$(u_i)_{i=1}^n \leftarrow (d_i)_{i=1}^n,\ (a_p)_{p\in P} \leftarrow 0,\ (s_t)_{t\in T} \leftarrow a(t),\ (r_p)_{p\in P} \leftarrow 0$

while $\exists i \in \{1,\dots,n\} : u_i > 0$ **do**

$$(a, p^{\text{best}}) \leftarrow \arg\min_{(a,p)\in\mathbb{N}\times P : s_{t(p)} \geq a > 0} \frac{\sum_{i=1}^n \max(a\cdot e_i^p, u_i)\cdot v_i}{a\cdot c_p^{\mathrm{P}} + \left\lceil \frac{a - r_p}{s^{\max}} \right\rceil \cdot c_p^{\mathrm{S}}}$$

$$a_{p^{\text{best}}} \leftarrow a_{p^{\text{best}}} + a$$

$$s_{t(p^{\text{best}})} \leftarrow s_{t(p^{\text{best}})} + a$$

$$r_{p^{\text{best}}} \leftarrow s^{\max} - 1 - (a_{p^{\text{best}}} - 1 \mod s^{\max})$$

$$u_i \leftarrow u_i - e_i^{p^{\text{best}}} \cdot a \quad \forall i = 1,\dots,n$$

end while

Extension Ideas

- Truncated Branch & Bound
- Stopping the greedy at some specified point and finish solution with construction heuristic
- Use greedy not only as post processing but also in the solution generation process itself