Optiblech

November 15, 2015

Problem: Optiblech

Eingabe

- $(dx, dy) \in \mathbb{R}^2$, Maße einer rechteckigen Blechtafel
- $\{((x_{1_1}, y_{1_1}), \ldots, (x_{1_{k(1)}}, y_{1_{k(1)}})), \ldots, ((x_{m_1}, y_{m_1}), \ldots, (x_{m_{k(m)}}, y_{m_{k(m)}}))\}$, Menge der m auszustanzenden Formen, gegeben durch die Punktetupel $((x_{i_1}, y_{i_1}), \ldots, (x_{i_{k(i)}}, y_{i_{k(i)}}))$ die sie definieren, wobei $k(i)$ angibt, wie viele Punkte die i-te Form hat
- $\{n_1, \ldots, n_m\} \in \mathbb{N}^m$, Menge in der n_i beschreibt wie oft eine Form $i \in \{1, \ldots, m\}$ ausgestanzt werden soll

Ausgabe

Sei, b die Anzahl der Bleche, A_b die Fläche der b Bleche, A_i die Fläche der i-ten Form und $A_f = \sum_{i=1}^{m} n_i \cdot A_i$ die Summe die Flächen aller ausgestanzten Formen. Ausgegeben wird dann b, die Anzahl der Bleche, die mindestens benötigt werden, um n_i mal die Form i auszustanzen, sodass $A_b - A_f$ möglichst klein ist und die Formen nicht überlappen, bzw. die Grenzen eines Blechs überschreiten.

Problem: Das Behälterproblem (Binpacking)

Eingabe

- $g \in \mathbb{N}$, die Größe der Behälter (Bins)
- $a_1, \ldots, a_n$, die Größe der n Objekte, mit $a_i < g$

Ausgabe

Die Anzahl b, der Behälter die mindestens benötigt werden, um die n Objekte in ihnen unterzubringen, ohne dass die Behälter “überlaufen” oder sich die Objekte überlappen.

Reduktion

Es ist bekannt, dass das Binpacking-Problem NP-schwer ist. Wir reduzieren Binpacking auf Optiblech um zu zeigen, dass auch Optiblech NP-schwer ist. Gegeben sei eine Instanz $(g, \{a_1, \ldots, a_n\})$ von Binpacking. Wir überführen diese in eine Optiblech-Instanz $((dx, dy), \{((x_{1_1}, y_{1_1}), \ldots, (x_{1_{k(1)}}, y_{1_{k(1)}})), \ldots, ((x_{m_1}, y_{m_1}), \ldots, (x_{m_{k(m)}}, y_{m_{k(m)}}))\}, \{n_1, \ldots, n_m\})$, wie folgt. Wähle als Blechgröße $(g, 1)$, wobei g die Größe der Bins ist. Nun werden die Objektgrößen in Punktmengen der Formen des Optiblechproblems überführt. Erstelle für das Objekt mit Größe a_i die Punktmenge $((0,0),(a_i,0),(a_i,1),(0,1))$. Die Menge die beschreibt, wie oft eine Form ausgestanzt werden soll, wird aus n $1en$ bestehen, da jedes Objekt genau einmal untergebracht werden soll. Diese Reduktion ist offensichtlich in polynomieller Zeit durchführbar.

Behauptung

Die minimale Anzahl an Blechen, die ein Algorithmus für Optiblech bei der resultierenden Instanz: $((g,1), \{((0,0),(a_1,0),(a_1,1),(0,1)), \ldots, ((0,0),(a_n,0),(a_n,1),(0,1))\}, \{1\}^n)$ ausgeben würde, ist auch die minimale Anzahl an Bins der Binpacking-Instanz.

Beweis

Da alle Formen die gleiche Höhe von 1 haben, kann in der Optiblech-Instanz nicht von Rotation profitiert werden. Formen die länger als 1 sind, können nicht gedreht werden, ohne die Grenzen des Bleches zu überschreiten. Bei Formen der Länge 1, würde eine Rotation keine Änderung bewirken. Formen mit Länge < 1 könnten nur dann von Rotation profitieren, wenn die verbleibende vertikale Fläche durch anderen Formen der Länge < 1 aufgefüllt werden kann. Da dies aber nur möglich ist, wenn die gemeinsam eingenommene Fläche dieser Formen exakt 1×1 ist, wäre die gleiche Effizienz auch ohne Rotation möglich gewesen. Rotation hat demzufolge keinen Einfluss auf die ausgenutzte Fläche im Optiblechproblem. Da alle Formen genauso hoch wie die Bleche sind und nicht rotiert werden können, kann auch die Positionierung der Formen auf einem Blech die Effizienz der ausgenutzten Fläche nicht beeinflussen. Lediglich die Zuordnung der Formen auf die verschiedenen Bleche beeinflusst diesen Faktor. Diese Aufteilung im Optiblechproblem ist auch im Binpackingproblem zulässig, da die Formen sich nicht überlappen oder die Blechgrenzen überschreiten dürfen. Demzufolge würde in der Optiblech-Instanz eine Aufteilung der Objekte gefunden werden, die die Anzahl der Bleche minimiert. Die gleiche Aufteilung der Objekte der Binpacking-Instanz würde dann auch die Anzahl der Bins minimieren. Somit könnte Binpacking mit Optiblech gelöst werden. Da Binpacking NP-schwer ist, folgt daraus, dass auch Optbiblech NP-schwer sein muss.