FORMAL MODELLING

Modelling Problems in Geometry and Discrete Mathematics

Wolfgang Windsteiger
Research Institute for Symbolic Computation (RISC)
Johannes Kepler University, Linz, Austria
Wolfgang.Windsteiger@risc.jku.at

JOHANNES KEPLER UNIVERSITY LINZ

RESEARCH INSTITUTE FOR SYMBOLIC COMPUTATION

MODELLING IN COMBINATORIAL OPTIMIZATION

COMBINATORIAL OPTIMIZATION

- Combinatorial optimization deals with the optimization of an objective function over a finite domain (e.g. integers or natural numbers).

COMBINATORIAL OPTIMIZATION

- Combinatorial optimization deals with the optimization of an objective function over a finite domain (e.g. integers or natural numbers).
- Restrictions allow only finitely many feasible solutions.

COMBINATORIAL OPTIMIZATION

- Combinatorial optimization deals with the optimization of an objective function over a finite domain (e.g. integers or natural numbers).
- Restrictions allow only finitely many feasible solutions.
- "Solution in principle": exhaustive search through all finitely many feasible solutions.

COMBINATORIAL OPTIMIZATION

- Combinatorial optimization deals with the optimization of an objective function over a finite domain (e.g. integers or natural numbers).
- Restrictions allow only finitely many feasible solutions.
- “Solution in principle”: exhaustive search through all finitely many feasible solutions.
- Examples: travelling salesman problem, minimum spanning tree problem, the knapsack problem, or the bin packing problem.

INTRODUCTORY EXAMPLES

Example

A factory has 10 production stations with equal capabilities. Each machine can be operated for at most 9 hours per day, production may start at 8:30. Every station needs two workers for operation, if a station stays closed the two employees can be used for other useful tasks. There are 160 orders with different production duration that have to be processed on a certain day. Each order can be processed on any of the stations. The delivery of the final products is scheduled on the night train leaving the factory no earlier than 18:00. Time for packing the products on the train is less than half an hour.

Design a “good” production schedule for that day.

INTRODUCTORY EXAMPLES

Example

An online shop delivers goods in boxes of maximum capacity 2kg. We have a concrete order with 96 items with known weights $w_1, \ldots, w_{96}$, respectively. How many boxes do we need to ship all ordered items such that the capacity restrictions are all satisfied?

INTRODUCTORY EXAMPLES

Example

An online shop delivers goods in boxes of maximum capacity 2kg. We have a concrete order with 96 items with known weights $w_1, \ldots, w_{96}$, respectively. How many boxes do we need to ship all ordered items such that the capacity restrictions are all satisfied?

Common pattern:

Distribute items (with given sizes)

INTRODUCTORY EXAMPLES

Example

An online shop delivers goods in boxes of maximum capacity 2kg. We have a concrete order with 96 items with known weights $w_1, \ldots, w_{96}$, respectively. How many boxes do we need to ship all ordered items such that the capacity restrictions are all satisfied?

Common pattern:

Distribute items (with given sizes) to boxes (with given capacities).

THE BIN PACKING PROBLEM

The pattern described above is known in literature as the bin packing problem.

THE BIN PACKING PROBLEM

The pattern described above is known in literature as the bin packing problem.

***BPP*(a, A): Bin Packing Problem**

Given: Positive numbers $a_1, \dots, a_n, A$.

Find: $k \in \mathbb{N}$ and $p : \mathbb{N}_{1,n} \to \mathbb{N}_{1,k}$ such that $\underset{1 \le j \le k}{\forall} \sum_{\substack{i \\ p(i)=j}} a_i \le A$.

- Given n items $I_1, \dots, I_n$ we need to find a number k of bins $B_1, \dots, B_k$.

THE BIN PACKING PROBLEM

The pattern described above is known in literature as the bin packing problem.

BPP(a, A): Bin Packing Problem

Given: Positive numbers $a_1, \ldots, a_n, A$.

Find: $k \in \mathbb{N}$ and $p : \mathbb{N}_{1,n} \to \mathbb{N}_{1,k}$ such that $\underset{1 \le j \le k}{\forall} \sum_{\substack{i \\ p(i)=j}} a_i \le A$.

- Given n items $I_1, \ldots, I_n$ we need to find a number k of bins $B_1, \ldots, B_k$.
- Assignment through function p, which assigns item index i a bin index j.

THE BIN PACKING PROBLEM

The pattern described above is known in literature as the bin packing problem.

$BPP(a, A)$: Bin Packing Problem

Given: Positive numbers $a_1, \ldots, a_n, A$.

Find: $k \in \mathbb{N}$ and $p \colon \mathbb{N}_{1,n} \to \mathbb{N}_{1,k}$ such that $\underset{1 \le j \le k}{\forall} \sum_{\substack{i \\ p(i)=j}} a_i \le A$.

- Given n items $I_1, \ldots, I_n$ we need to find a number k of bins $B_1, \ldots, B_k$.
- Assignment through function p, which assigns item index i a bin index j.
- $p(i) = j$ means: item I_i is packed into bin B_j.

THE BIN PACKING PROBLEM

The pattern described above is known in literature as the bin packing problem.

BPP(a, A): Bin Packing Problem

Given: Positive numbers $a_1, \ldots, a_n, A$.

Find: $k \in \mathbb{N}$ and $p : \mathbb{N}_{1,n} \to \mathbb{N}_{1,k}$ such that $\underset{1 \le j \le k}{\forall} \sum_{\substack{i \\ p(i)=j}} a_i \le A$.

- Given n items $I_1, \ldots, I_n$ we need to find a number k of bins $B_1, \ldots, B_k$.
- Assignment through function p, which assigns item index i a bin index j.
- $p(i) = j$ means: item I_i is packed into bin B_j.
- (k, p) is a feasible solution for *BPP*(a, A) if they satisfy the above conditions.

VARIANTS OF THE BIN PACKING PROBLEM I

BPDP(a, A, k): Bin Packing Decision Problem

Given: Positive numbers $a_1, \ldots, a_n, A$ and $k \in \mathbb{N}$.

Question: Does there exist a function $p : \mathbb{N}_{1,n} \to \mathbb{N}_{1,k}$ such that (k, p) a feasible solution for *BPP*(a, A)?

VARIANTS OF THE BIN PACKING PROBLEM II

BPOP(a, A): Bin Packing Optimization Problem

Given: Positive numbers $a_1, \ldots, a_n, A$.

Find: $k \in \mathbb{N}$ and $p \colon \mathbb{N}_{1,n} \to \mathbb{N}_{1,k}$ such that

1. (k, p) a feasible solution for *BPP*(a, A) and
2. k is minimal, i.e.

$$\underset{m<k}{\forall} \quad \underset{q \colon \mathbb{N}_{1,n} \to \mathbb{N}_{1,m}}{\forall} (m, q) \text{ is not a feasible solution for } \mathit{BPP}(a, A).$$

BIN PACKING AS LINEAR PROGRAMMING PROBLEM

$$\alpha_{ij} := \begin{cases} 1 & \text{item } I_i \text{ goes into bin } B_j \\ 0 & \text{otherwise} \end{cases} \qquad \beta_j := \begin{cases} 1 & \text{bin } B_j \text{ will be occupied} \\ 0 & \text{otherwise} \end{cases}$$

BIN PACKING AS LINEAR PROGRAMMING PROBLEM

$$\alpha_{ij} := \begin{cases} 1 & \text{item } I_i \text{ goes into bin } B_j \\ 0 & \text{otherwise} \end{cases} \qquad \beta_j := \begin{cases} 1 & \text{bin } B_j \text{ will be occupied} \\ 0 & \text{otherwise} \end{cases}$$

BPLPP(a, A) :

$$\forall_{1 \le j \le n} \sum_{i=1}^{n} \alpha_{ij} a_i \le A\beta_j \qquad \text{(capacity per bin)}$$

BIN PACKING AS LINEAR PROGRAMMING PROBLEM

$$\alpha_{ij} := \begin{cases} 1 & \text{item } I_i \text{ goes into bin } B_j \\ 0 & \text{otherwise} \end{cases} \qquad \beta_j := \begin{cases} 1 & \text{bin } B_j \text{ will be occupied} \\ 0 & \text{otherwise} \end{cases}$$

BPLPP(a, A) :

$$\forall_{1 \le j \le n} \sum_{i=1}^{n} \alpha_{ij} a_i \le A\beta_j \qquad \text{(capacity per bin)}$$

$$\forall_{1 \le i \le n} \sum_{j=1}^{n} \alpha_{ij} = 1, \qquad \text{(every item in exactly one bin)}$$

BIN PACKING AS LINEAR PROGRAMMING PROBLEM

$$\alpha_{ij} := \begin{cases} 1 & \text{item } I_i \text{ goes into bin } B_j \\ 0 & \text{otherwise} \end{cases} \qquad \beta_j := \begin{cases} 1 & \text{bin } B_j \text{ will be occupied} \\ 0 & \text{otherwise} \end{cases}$$

BPLPP(a, A) :

$$\underset{1 \le j \le n}{\forall} \sum_{i=1}^{n} \alpha_{ij} a_i \le A\beta_j \qquad \text{(capacity per bin)}$$

$$\underset{1 \le i \le n}{\forall} \sum_{j=1}^{n} \alpha_{ij} = 1, \qquad \text{(every item in exactly one bin)}$$

$$\sum_{j=1}^{n} \beta_j \longrightarrow \text{Min} \qquad \text{(minimum number of bins)}$$

BIN PACKING AS LINEAR PROGRAMMING PROBLEM

$$\alpha_{ij} := \begin{cases} 1 & \text{item } I_i \text{ goes into bin } B_j \\ 0 & \text{otherwise} \end{cases} \qquad \beta_j := \begin{cases} 1 & \text{bin } B_j \text{ will be occupied} \\ 0 & \text{otherwise} \end{cases}$$

$$\mathit{BPLPP}(a, A) : \qquad \underset{1 \le j \le n}{\forall} \sum_{i=1}^{n} \alpha_{ij} a_i \le A\beta_j \qquad \text{(capacity per bin)}$$

$$\underset{1 \le i \le n}{\forall} \sum_{j=1}^{n} \alpha_{ij} = 1, \qquad \text{(every item in exactly one bin)}$$

$$\sum_{j=1}^{n} \beta_j \longrightarrow \text{Min} \qquad \text{(minimum number of bins)}$$

$\mathit{BPLPP}(a, A)$ forms an integer linear programming problem with integer variables

$$\alpha_{ij} \in \{0, 1\} \text{ for } 1 \le i, j \le n \quad \text{and} \quad \beta_j \in \{0, 1\} \text{ for } 1 \le j \le n.$$

LINEAR PROGRAMMING STANDARD FORM

Standard form of a linear programming problem for variables $x \in \mathbb{R}^t$:

$M \cdot x \equiv b$	with $M \in \mathbb{R}^{s \times t}, b \in \mathbb{R}^s, \equiv \in \{\leq, =, \geq\}$
$c \cdot x \longrightarrow$ Min	with $c \in \mathbb{R}^t$

LINEAR PROGRAMMING STANDARD FORM

Standard form of a linear programming problem for variables $x \in \mathbb{R}^t$:

$$M \cdot x \equiv b \qquad \text{with } M \in \mathbb{R}^{s \times t}, b \in \mathbb{R}^s, \equiv\, \in \{\leq, =, \geq\}$$

$$c \cdot x \longrightarrow \text{Min} \qquad \text{with } c \in \mathbb{R}^t$$

For the bin packing problem, the setting from above results in $x \in \mathbb{R}^{n^2+n}$ with

$$x_v := \begin{cases} \alpha_{q+1,r+1} & 1 \leq v \leq n^2, (q,r) = \text{QuotRem}(v-1,n) \\ \beta_{v-n^2} & n^2+1 \leq v \leq n^2+n. \end{cases}$$

LINEAR PROGRAMMING STANDARD FORM

Standard form of a linear programming problem for variables $x \in \mathbb{R}^t$:

$$M \cdot x \equiv b \qquad \text{with } M \in \mathbb{R}^{s \times t}, b \in \mathbb{R}^s, \equiv \in \{\leq, =, \geq\}$$

$$c \cdot x \longrightarrow \text{Min} \qquad \text{with } c \in \mathbb{R}^t$$

For the bin packing problem, the setting from above results in $x \in \mathbb{R}^{n^2+n}$ with

$$x_v := \begin{cases} \alpha_{q+1,r+1} & 1 \leq v \leq n^2, (q,r) = \text{QuotRem}(v-1, n) \\ \beta_{v-n^2} & n^2+1 \leq v \leq n^2+n. \end{cases}$$

Since $v = nq + r + 1$ we get

$$\alpha_{ij} = x_{n(i-1)+j} \qquad \beta_j = x_{n^2+j}.$$

THE MATRIX $M \in \mathbb{R}^{(2n)\times(n^2+n)}$

Rows 1–n: Capacity restrictions for $1 \le j \le n$

$$(M \cdot x)_j = \sum_{\nu=1}^{n^2+n} M_{j\nu} x_\nu = \sum_{i=1}^{n} \alpha_{ij} a_i - A\beta_j = \sum_{i=1}^{n} x_{n(i-1)+j} a_i - A x_{n^2+j}.$$

THE MATRIX $M \in \mathbb{R}^{(2n)\times(n^2+n)}$

Rows 1–n: Capacity restrictions for $1 \leq j \leq n$

$$(M \cdot x)_j = \sum_{v=1}^{n^2+n} M_{jv} x_v = \sum_{i=1}^{n} \alpha_{ij} a_i - A\beta_j = \sum_{i=1}^{n} x_{n(i-1)+j} a_i - A x_{n^2+j}.$$

Rows $n+1$–$2n$: Unicity restrictions for $1 \leq i \leq n$

$$(M \cdot x)_{n+i} = \sum_{v=1}^{n^2+n} M_{n+i,v} x_v = \sum_{j=1}^{n} \alpha_{ij} = \sum_{j=1}^{n} x_{n(i-1)+j}.$$

THE MATRIX M: CAPACITY RESTRICTIONS

$$\text{for } 1 \le j \le n: \quad (M \cdot x)_j = \underbrace{\sum_{\nu=1}^{n^2+n} M_{j\nu} x_\nu}_{(*)} = \sum_{i=1}^{n} \alpha_{ij} a_i - A\beta_j = \underbrace{\sum_{i=1}^{n} x_{n(i-1)+j} a_i - A x_{n^2+j}}_{(**)} .$$

Compare the coefficients $M_{j\nu}$ of x_ν in $(*)$ with those in $(**)$:

- If $\nu = n(i-1) + j$ for some $1 \le i \le n$ then the coefficient is $M_{j\nu} = a_i$. Note, that the condition is equivalent to $\nu \bmod n = j \bmod n$.

THE MATRIX M: CAPACITY RESTRICTIONS

$$\text{for } 1 \leq j \leq n: \quad (M \cdot x)_j = \underbrace{\sum_{v=1}^{n^2+n} M_{jv} x_v}_{(*)} = \sum_{i=1}^{n} \alpha_{ij} a_i - A\beta_j = \underbrace{\sum_{i=1}^{n} x_{n(i-1)+j} a_i - A x_{n^2+j}}_{(**)} .$$

Compare the coefficients M_{jv} of x_v in $(*)$ with those in $(**)$:

- If $v = n(i-1) + j$ for some $1 \leq i \leq n$ then the coefficient is $M_{jv} = a_i$. Note, that the condition is equivalent to $v \bmod n = j \bmod n$.
- If $v = n^2 + j$ then the coefficient is $M_{jv} = -A$.

THE MATRIX M: CAPACITY RESTRICTIONS

$$\text{for } 1 \leq j \leq n: \quad (M \cdot x)_j = \underbrace{\sum_{v=1}^{n^2+n} M_{jv} x_v}_{(*)} = \sum_{i=1}^{n} \alpha_{ij} a_i - A\beta_j = \underbrace{\sum_{i=1}^{n} x_{n(i-1)+j} a_i - A x_{n^2+j}}_{(**)} .$$

Compare the coefficients M_{jv} of x_v in $(*)$ with those in $(**)$:

- If $v = n(i-1) + j$ for some $1 \leq i \leq n$ then the coefficient is $M_{jv} = a_i$. Note, that the condition is equivalent to $v \bmod n = j \bmod n$.
- If $v = n^2 + j$ then the coefficient is $M_{jv} = -A$.
- For all other coefficients we have $M_{jv} = 0$.

THE MATRIX M: UNICITY RESTRICTIONS

$$\text{for } 1 \leq i \leq n: \quad (M \cdot x)_{n+i} = \underbrace{\sum_{v=1}^{n^2+n} M_{n+i,v} x_v}_{(*)} = \sum_{j=1}^{n} \alpha_{ij} = \underbrace{\sum_{j=1}^{n} x_{n(i-1)+j}}_{(**)} .$$

Compare the coefficients $M_{n+i,v}$ of x_v in $(*)$ with those in $(**)$:

- If $n(i-1)+1 \leq v \leq ni$ then the coefficient is $M_{n+i,v} = 1$.

THE MATRIX M: UNICITY RESTRICTIONS

$$\text{for } 1 \leq i \leq n: \qquad (M \cdot x)_{n+i} = \underbrace{\sum_{v=1}^{n^2+n} M_{n+i,v} x_v}_{(*)} = \sum_{j=1}^{n} \alpha_{ij} = \underbrace{\sum_{j=1}^{n} x_{n(i-1)+j}}_{(**)} .$$

Compare the coefficients $M_{n+i,v}$ of x_v in $(*)$ with those in $(**)$:

- If $n(i-1)+1 \leq v \leq ni$ then the coefficient is $M_{n+i,v} = 1$.
- For all other coefficients we have $M_{n+i,v} = 0$.

THE MATRIX M AND THE RIGHT-HAND SIDE b

$$M_{jv} := \begin{cases} a_{\frac{v-j}{n}+1} & 1 \leq j \leq n \wedge (v \bmod n = j \bmod n) \wedge v \leq n^2 \\ -A & 1 \leq j \leq n \wedge v = n^2 + j \\ 1 & j > n \wedge n(j-n-1)+1 \leq v \leq n(j-n) \\ 0 & \text{otherwise.} \end{cases}$$

THE MATRIX M AND THE RIGHT-HAND SIDE b

$$M_{jv} := \begin{cases} a_{\frac{v-j}{n}+1} & 1 \le j \le n \wedge (v \bmod n = j \bmod n) \wedge v \le n^2 \\ -A & 1 \le j \le n \wedge v = n^2 + j \\ 1 & j > n \wedge n(j-n-1)+1 \le v \le n(j-n) \\ 0 & \text{otherwise.} \end{cases}$$

$$b \in \mathbb{R}^{2n} \text{ with } b_j := \begin{cases} 0 & 1 \le j \le n \\ 1 & \text{otherwise.} \end{cases}$$

THE MATRIX M AND THE RIGHT-HAND SIDE b

$$M_{jv} := \begin{cases} a_{\frac{v-j}{n}+1} & 1 \le j \le n \wedge (v \bmod n = j \bmod n) \wedge v \le n^2 \\ -A & 1 \le j \le n \wedge v = n^2 + j \\ 1 & j > n \wedge n(j-n-1)+1 \le v \le n(j-n) \\ 0 & \text{otherwise.} \end{cases}$$

$$b \in \mathbb{R}^{2n} \text{ with } b_j := \begin{cases} 0 & 1 \le j \le n \\ 1 & \text{otherwise.} \end{cases}$$

The restrictions $(M \cdot x)_j \equiv b_j$: $\quad \equiv := \begin{cases} \le & 1 \le j \le n \\ = & \text{otherwise.} \end{cases}$

THE OBJECTIVE FUNCTION $c \in \mathbb{R}^{n^2+n}$

$$c \cdot x = \underbrace{\sum_{\nu=1}^{n^2+n} c_\nu x_\nu}_{(*)} = \sum_{j=1}^{n} \beta_j = \underbrace{\sum_{j=1}^{n} x_{n^2+j}}_{(**)} .$$

Compare the coefficients of x_ν in $(*)$ with those in $(**)$:

THE OBJECTIVE FUNCTION $c \in \mathbb{R}^{n^2+n}$

$$c \cdot x = \underbrace{\sum_{\nu=1}^{n^2+n} c_\nu x_\nu}_{(*)} = \sum_{j=1}^{n} \beta_j = \underbrace{\sum_{j=1}^{n} x_{n^2+j}}_{(**)} .$$

Compare the coefficients of x_ν in $(*)$ with those in $(**)$:

$$c_\nu := \begin{cases} 0 & 1 \le \nu \le n^2 \\ 1 & \text{otherwise.} \end{cases}$$

BIN PACKING: SOLUTIONS

Suppose (α, β) a solution of *BPLPP*(a, A). Let $\pi : \mathbb{N}_{1,n} \to \mathbb{N}_{1,n}$ be a permutation s.t.

$$\underset{1 \le j \le k}{\forall} \beta_{\pi(j)} = 1 \wedge \underset{k < j \le n}{\forall} \beta_{\pi(j)} = 0.$$

π permutes the bins: bin j occupied if and only if $\beta_{\pi(j)} = 1$.

BIN PACKING: SOLUTIONS

Suppose (α, β) a solution of *BPLPP*(a, A). Let $\pi : \mathbb{N}_{1,n} \to \mathbb{N}_{1,n}$ be a permutation s.t.

$$\underset{1 \leq j \leq k}{\forall} \beta_{\pi(j)} = 1 \wedge \underset{k < j \leq n}{\forall} \beta_{\pi(j)} = 0.$$

π permutes the bins: bin j occupied if and only if $\beta_{\pi(j)} = 1$. Then

$$k := \sum_{j=1}^{n} \beta_j \qquad\qquad p : \mathbb{N}_{1,n} \to \mathbb{N}_{1,k}, i \mapsto \text{the unique } j \text{ with } \alpha_{i\,\pi(j)} = 1$$

is a solution of *BPOP*(a, A).

BIN PACKING: CORRECTNESS

$$k := \sum_{j=1}^{n} \beta_j \qquad\qquad p : \mathbb{N}_{1,n} \to \mathbb{N}_{1,k}, i \mapsto \text{the unique } j \text{ with } \alpha_{i\pi(j)} = 1$$

1. p is well-defined: for every $1 \le i \le n$ the unique existence of j follows immediately from $\sum_{j=1}^{n} \alpha_{i\pi(j)} = \sum_{j=1}^{n} \alpha_{ij} = 1$ together with $\alpha_{i\pi(j)} \in \{0,1\}$.

BIN PACKING: CORRECTNESS

$$k := \sum_{j=1}^{n} \beta_j \qquad p\colon \mathbb{N}_{1,n} \to \mathbb{N}_{1,k}, i \mapsto \text{the unique } j \text{ with } \alpha_{i\pi(j)} = 1$$

1. p is well-defined: for every $1 \leq i \leq n$ the unique existence of j follows immediately from $\sum_{j=1}^{n} \alpha_{i\pi(j)} = \sum_{j=1}^{n} \alpha_{ij} = 1$ together with $\alpha_{i\pi(j)} \in \{0,1\}$.
2. (k,p) is feasible: let $1 \leq j \leq k$ arbitrary but fixed and now

$$\sum_{\substack{i \\ p(i)=j}} a_i = \sum_{\substack{i \\ \alpha_{i\pi(j)}=1}} a_i = \sum_{i=1}^{n} \alpha_{i\pi(j)} a_i \overset{\text{BPLPP}}{\leq} A\beta_{\pi(j)} \overset{1\leq j\leq k}{=} A.$$

BIN PACKING: CORRECTNESS

$$k := \sum_{j=1}^{n} \beta_j \qquad\qquad p \colon \mathbb{N}_{1,n} \to \mathbb{N}_{1,k}, i \mapsto \text{the unique } j \text{ with } \alpha_{i\pi(j)} = 1$$

1. p is well-defined: for every $1 \le i \le n$ the unique existence of j follows immediately from $\sum_{j=1}^{n} \alpha_{i\pi(j)} = \sum_{j=1}^{n} \alpha_{ij} = 1$ together with $\alpha_{i\pi(j)} \in \{0, 1\}$.
2. (k, p) is feasible: let $1 \le j \le k$ arbitrary but fixed and now

$$\sum_{\substack{i \\ p(i)=j}} a_i = \sum_{\substack{i \\ \alpha_{i\pi(j)}=1}} a_i = \sum_{i=1}^{n} \alpha_{i\pi(j)} a_i \overset{\textit{BPLPP}}{\le} A\beta_{\pi(j)} \overset{1\le j\le k}{=} A.$$

3. k is minimal because $k = \sum_{j=1}^{n} \beta_j$ together with *BPLPP*.

EXAMPLE

See Mathematica-Demo.

APPROXIMATION ALGORITHMS: TERMINOLOGY

- $P(x)$ ... instance of problem P with input x.

APPROXIMATION ALGORITHMS: TERMINOLOGY

- $P(x) \ldots$ instance of problem P with input x.
- $A(x) \ldots$ result of algorithm A applied to x.

APPROXIMATION ALGORITHMS: TERMINOLOGY

- $P(x)$ ... instance of problem P with input x.
- $A(x)$ ... result of algorithm A applied to x.
- Structure of optimization problems P: Feasibility + Optimality.

APPROXIMATION ALGORITHMS: TERMINOLOGY

- $P(x)$ ... instance of problem P with input x.
- $A(x)$ ... result of algorithm A applied to x.
- Structure of optimization problems P: Feasibility + Optimality.
- Given x, find y s.t. ...

APPROXIMATION ALGORITHMS: TERMINOLOGY

- $P(x)$ ... instance of problem P with input x.
- $A(x)$ ... result of algorithm A applied to x.
- Structure of optimization problems P: Feasibility + Optimality.
- Given x, find y s.t. ...
  - Feasibility: y fulfills $\bar{P}$.

APPROXIMATION ALGORITHMS: TERMINOLOGY

- $P(x)$ ... instance of problem P with input x.
- $A(x)$ ... result of algorithm A applied to x.
- Structure of optimization problems P: Feasibility + Optimality.
- Given x, find y s.t. ...
  - Feasibility: y fulfills $\bar{P}$.
  - Optimality: y is minimal/maximal w.r.t. some measure.

APPROXIMATION ALGORITHMS: TERMINOLOGY

- $P(x) \ldots$ instance of problem P with input x.
- $A(x) \ldots$ result of algorithm A applied to x.
- Structure of optimization problems P: Feasibility + Optimality.
- Given x, find y s.t. $\ldots$
  - Feasibility: y fulfills $\bar{P}$.
  - Optimality: y is minimal/maximal w.r.t. some measure.
- For every optimization problem P there is the relaxed problem $\bar{P}$, which only covers feasibility but neglects optimality.

APPROXIMATION ALGORITHMS: TERMINOLOGY

Definition (Approximation Algorithm)

Let P be an optimization problem and $\bar{P}$ the relaxed problem ignoring optimality. We call A an approximation algorithm for problem P if and only if every $y = A(x)$ is still a solution of $\bar{P}(x)$ but not necessarily a solution of $P(x)$.

APPROXIMATION ALGORITHMS: TERMINOLOGY

Definition (Approximation Algorithm)

Let P be an optimization problem and $\bar{P}$ the relaxed problem ignoring optimality. We call A an approximation algorithm for problem P if and only if every $y = A(x)$ is still a solution of $\bar{P}(x)$ but not necessarily a solution of $P(x)$.

Definition (Approximation Quality)

Let P be an optimization problem and $y(x)$ a solution for $P(x)$. We call A a k-approximation algorithm ($k \geq 1$) for P iff for all admissible inputs x of P

$$A(x) \begin{cases} \leq k y(x) & \text{if } P \text{ is a minimization problem} \\ \geq \frac{1}{k} y(x) & \text{if } P \text{ is a maximization problem} \end{cases}$$

HEURISTIC APPROXIMATION ALGORITHMS FOR BPP

Theorem

If $P \neq NP$, then there is no k-approximation algorithm for the optimal bin packing problem with $k < \frac{3}{2}$.

HEURISTIC APPROXIMATION ALGORITHMS FOR BPP

Theorem

If $P \neq NP$, then there is no k-approximation algorithm for the optimal bin packing problem with $k < \frac{3}{2}$.

Theorem

FFD is a $\frac{3}{2}$-approximation algorithm for the optimal bin packing problem.

HEURISTIC APPROXIMATION ALGORITHMS FOR BPP

Theorem

If $P \neq NP$, then there is no k-approximation algorithm for the optimal bin packing problem with $k < \frac{3}{2}$.

Theorem

FFD is a $\frac{3}{2}$-approximation algorithm for the optimal bin packing problem.

Theorem

For all instances x of the bin packing problem with solution $y(x)$ we have

$$\mathit{FFD}(x) \leq \frac{11}{9} y(x) + \frac{2}{3}.$$

FFD: FIRST FIT DECREASING HEURISTIC FOR BPP

- Simple heuristic for solving BPP.

FFD: FIRST FIT DECREASING HEURISTIC FOR BPP

- Simple heuristic for solving BPP.
- Sort items by decreasing size (big items first).

FFD: FIRST FIT DECREASING HEURISTIC FOR BPP

- Simple heuristic for solving BPP.
- Sort items by decreasing size (big items first).
- For every item: go through all occupied bins and put item in first possible bin.

FFD: FIRST FIT DECREASING HEURISTIC FOR BPP

- Simple heuristic for solving BPP.
- Sort items by decreasing size (big items first).
- For every item: go through all occupied bins and put item in first possible bin.
- If it does not fit in any, then open a new bin.

EXAMPLES

See Mathematica Demo.