

Packing under Convex Quadratic Constraints ^{*}

Max Klimm¹, Marc E. Pfetsch², Rico Raber³, and Martin Skutella³

¹ School of Business and Economics, HU Berlin, Spandauer Str. 1, 10178 Berlin, Germany, max.klimm@hu-berlin.de.

² Department of Mathematics, TU Darmstadt, Dolivostr. 15, 64293 Darmstadt, Germany, pfetsch@mathematik.tu-darmstadt.de

³ Institute of Mathematics, TU Berlin, Straße des 17. Juni 136, 10623 Berlin, Germany, {raber,skutella}@math.tu-berlin.de

Abstract. We consider a general class of binary packing problems with a convex quadratic knapsack constraint. We prove that these problems are APX-hard to approximate and present constant-factor approximation algorithms based upon three different algorithmic techniques: (1) a rounding technique tailored to a convex relaxation in conjunction with a non-convex relaxation whose approximation ratio equals the golden ratio; (2) a greedy strategy; (3) a randomized rounding method leading to an approximation algorithm for the more general case with multiple convex quadratic constraints. We further show that a combination of the first two strategies can be used to yield a monotone algorithm leading to a strategyproof mechanism for a game-theoretic variant of the problem. Finally, we present a computational study of the empirical approximation of the three algorithms for problem instances arising in the context of real-world gas transport networks.

1 Introduction

We consider packing problems with a convex quadratic knapsack constraint of the form

$$\begin{aligned} & \text{maximize} && p^\top x \\ & \text{subject to} && x^\top W x \leq c, \\ & && x \in \{0, 1\}^n, \end{aligned} \tag{P}$$

where $W \in \mathbb{Q}_{\geq 0}^{n \times n}$ is a symmetric positive semi-definite (psd) matrix with non-negative entries, $p \in \mathbb{Q}_{\geq 0}^n$ is a non-negative profit vector, and $c \in \mathbb{Q}_{\geq 0}$ is a non-negative budget. Such convex and quadratically constrained packing problems are clearly NP-complete since they contain the classical (linearly constrained) NP-complete knapsack problem [14] as a special case when W is a diagonal matrix. In this paper, we therefore focus on the development of approximation algorithms. For some $\rho \in [0, 1]$, an algorithm is a ρ -approximation algorithm if its runtime is polynomial in the input size and for every instance, it computes

^{*} We acknowledge funding through the DFG CRC/TRR 154, Subproject A007.

a solution with objective value at least ρ times that of an optimum solution. The value ρ is then called the *approximation ratio* of the algorithm. We note that the assumption on W being psd is necessary in order to allow for sensible approximation. To see this, observe that when W is the adjacency matrix of an undirected graph and $c = 0$, (P) encodes the problem of finding an independent set of maximal weight, which is NP-hard to approximate within a factor better than $n^{-(1-\epsilon)}$ for any $\epsilon > 0$, even in the unweighted case [10].

The packing problems that we consider also have a natural interpretation in terms of mechanism design. Consider a situation where a set of n selfish agents demands a service, and the subsets of agents that can be served simultaneously are modeled by a convex quadratic packing constraint. Each agent j has private information p_j about its willingness to pay for receiving the service. In this context, a (direct revelation) mechanism takes as input the matrix W and the budget c . It then elicits the private value p_j from agent j . Each agent j may misreport a value p'_j instead of their true value p_j if this is to their benefit. The mechanism then computes a solution $x \in \{0, 1\}^n$ to (P) as well as a payment vector $g \in \mathbb{Q}_{\geq 0}^n$. A mechanism is strategyproof if no agent has an interest in misreporting p_j , no matter what the other agents report.

Before we present our results on approximation ratios and mechanisms for non-negative, convex, and quadratically constrained packing problems, we give two real-world examples that fall into this category.

Example 1 (Welfare maximization in gas supply networks). Consider a gas pipeline modeled by a directed graph $G = (V, E)$ with different entry and exit nodes. There is a set of n transportation requests (s_j, t_j, q_j, p_j) , $j \in [n] := \{1, \dots, n\}$, each specifying an entry node $s_j \in V$, an exit node $t_j \in V$, the amount of gas to be transported $q_j \in \mathbb{Q}_{\geq 0}$, and an economic value $p_j \in \mathbb{Q}_{\geq 0}$. One model for gas flows in pipe networks is given by the Weymouth equations [28] of the form

$$\beta_e q_e |q_e| = \pi_u - \pi_v \quad \text{for all } e = (u, v) \in E.$$

Here, the parameter $\beta_e \in \mathbb{Q}_{> 0}$ is a pipe specific value that depends on physical properties of the pipe segment modeled by the edge, such as length, diameter, and roughness. Positive flow values $q_e > 0$ denote flow from u to v , while a negative q_e indicates flow in the opposite direction. The value π_u denotes the square of the pressure at node $u \in V$. In real-life gas networks, there is typically a bound $c \in \mathbb{Q}_{\geq 0}$ on the maximal difference of the squared pressures in the network. For the operation of gas networks, it is a natural problem to find the welfare-maximal subset of transportation requests that can be satisfied simultaneously while satisfying the pressure constraint.

To illustrate this problem, we consider the particular case in which the network has a path topology similar to the one depicted in Figure 1. We assume that for each request the entry node is left of the exit node. Thus, the pressure in the pipe is decreasing from left to right. For $j \in [n]$, let $E_j \subseteq E$ denote the set of edges on the unique (s_j, t_j) -path in G . Indexing the vertices v_0, \dots, v_k and edges e_1, \dots, e_k from left to right, the maximal squared pressure difference in

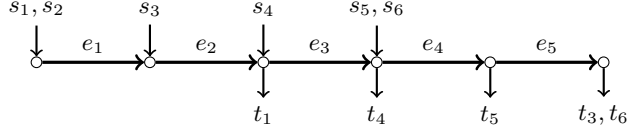


Fig. 1. Gas network with feed-in and feed-out nodes.

the pipe is given by

$$\pi_0 - \pi_k = \sum_{i=1}^k (\pi_{i-1} - \pi_i) = \sum_{i=1}^k \beta_{e_i} \left(\sum_{j \in [n]: e_i \in E_j} q_j x_j \right)^2,$$

where $x_j \in \{0, 1\}$ indicates whether transportation request $j \in [n]$ is being served. For the matrix $W = (w_{ij})_{i,j \in [n]}$ defined by $w_{ij} = \sum_{e \in E_i \cap E_j} \beta_e q_i q_j$, the pressure constraint can be formulated as $x^\top W x \leq c$. To see that the matrix W is positive semi-definite, we write $W = \sum_{e \in E} \beta_e q^e (q^e)^\top$, where $q^e \in \mathbb{Q}_{\geq 0}^n$ is defined as $q_i^e = q_i$ if $e \in E_i$, and $q_i^e = 0$, otherwise.

Gas networks are particularly interesting from a mechanism design perspective, since several countries employ or plan to employ auctions to allocate gas network capacities [21], but theoretical and experimental work uses only linear flow models [17,24], thus ignoring the physics of the gas flow.

Example 2 (Processor speed scaling). Consider a mobile device with battery capacity c and k compute cores. Further, there is a set of n tasks (q^j, p_j) , each specifying a load $q^j \in \mathbb{Q}_{\geq 0}^k$ for the k cores and a profit p_j . The computations start at time 0 and all computations have to be finished at time 1. In order to adapt to varying workloads, the compute cores can run at different speeds. In the speed scaling literature, it is a common assumption that energy consumption of core i when running at speed s is equal to $\beta_i s^2$, where $\beta_i \in \mathbb{Q}_{>0}$ is a core-specific parameter [2,13,29].¹ The goal is to select a profit-maximal subset of tasks that can be scheduled in the available time with the available battery capacity. Given a subset of tasks, it is without loss of generality to assume that each core runs at the minimal speed such that the core finishes at time 1, i.e., every core i runs at speed $\sum_{j \in [n]} x_j q_i^j$ so that the total energy consumption is $\sum_{i=1}^k \beta_i (\sum_{j \in [n]} x_j q_i^j)^2$. The energy constraint can thus be formulated as a convex quadratic constraint.

Mechanism design problems for processor speed scaling are interesting when the tasks are controlled by selfish agents and access to computation on the energy-constrained device is determined via an auction.

¹ Other works assume that the relationship is cubic, but experiments conducted by Wierman et al. [29] suggest that the relationship is closer to quadratic than cubic.

1.1 Our Results

In Section 3 we derive a ϕ -approximation algorithm for packing problems with convex quadratic constraints where $\phi = (\sqrt{5}-1)/2 \approx 0.618$ is the inverse golden ratio. The algorithm first solves a convex relaxation and scales the solution by ϕ , which turns it into a feasible solution to a second non-convex relaxation. The latter relaxation has the property that any solution can be transformed into a solution with at most one fractional component without decreasing the objective value. In the end, the algorithm returns the integral part of the transformed solution. Combining this procedure with a partial enumeration scheme yields a ϕ -approximation; see Theorem 1. In Section 4 we prove that the greedy algorithm, when combined with partial enumeration, is a constant-factor approximation algorithm with an approximation ratio between $(1 - \sqrt{3}/e) \approx 0.363$ and ϕ ; see Theorem 2 and Theorem 3. In Section 5, we show that a combination of the results from the previous section allows to derive a strategyproof mechanism with constant approximation ratio. In Section 6 we derive a randomized constant-factor approximation algorithm for the more general problem with a constant number of r convex quadratic packing constraints. The algorithm solves a convex relaxation, scales the solution, and performs randomized rounding based on that scaled solution. Combining this algorithm with partial enumeration yields a constant-factor approximation; see Theorem 5. In Section 7 we show that packing problems with convex quadratic constraints of type (P) are APX-hard; see Theorem 6. Finally, in Section 8, we apply the three algorithms to several instances of the problem type described in Example 1 based on real-world data from the GasLib library [26].

1.2 Related Work

When W is a non-negative diagonal matrix, the quadratic constraint in (P) becomes linear and the problem is then equivalent to the 0-1-knapsack problem which admits a *fully polynomial-time approximation scheme* (FPTAS) [12]. Another interesting special case is when W is completely-positive, i.e., it can then be written as $W = UU^\top$ for some matrix $U \in \mathbb{Q}_{\geq 0}^{n \times k}$ with non-negative entries. The minimal k for which W can be expressed in this way is called the *cp-rank* of W , see [3] for an overview on completely positive matrices. The quadratic constraint in (P) can then be expressed as $\|U^\top x\|_2 \leq \sqrt{c}$. For the case that $U \in \mathbb{Q}_{\geq 0}^{n \times 2}$, this problem is known as the 2-weighted knapsack problem for which Woeginger [30] showed that it does not admit an FPTAS, unless $\text{P} = \text{NP}$. Chau et al. [5] settled the complexity of this problem showing that it admits a *polynomial-time approximation scheme* (PTAS). Elbassioni et al. [6] generalized this result to matrices with constant cp-rank.

Exchanging constraints and objective in (P) leads to knapsack problems with quadratic objective function and a linear constraint first studied by Gallo [8]. These problems have a natural graph-theoretic interpretation where nodes and edges have profits, the nodes have weights, and the task is to choose a subset of nodes so as to maximize the total profit of the induced subgraph. Rader and

Woeginger [23] give an FPTAS when the graph is edge series-parallel. Pferschy and Schauer [22] generalize this result to graphs of bounded treewidth. They also give a PTAS for graphs not including a forbidden minor which includes planar graphs.

Mechanism design problems with a knapsack constraint are contained as a special case when W is a diagonal matrix. For this special case, Mu'alem and Nisan [18] give a mechanism that is strategyproof and yields a $1/2$ -approximation. Briest et al. [4] give a general framework that allows to construct a mechanism that is an FPTAS for the objective function. Aggarwal and Hartline [1] study knapsack auctions with the objective to maximize the sum of the payments to the mechanism.

2 Preliminaries

For ease of exposition, we assume that all matrices and vectors are integer. Let $[n] := \{1, \dots, n\}$ and $W = (w_{ij})_{i,j \in [n]} \in \mathbb{N}^{n \times n}$ be a symmetric psd matrix. Furthermore, let $p \in \mathbb{N}^n$ be a profit vector and let $c \in \mathbb{N}$ be a budget. We consider problems of the form (P) , i.e., $\max \{p^\top x : x^\top W x \leq c, x \in \{0, 1\}^n\}$. Throughout the paper, we denote the characteristic vector of a subset $S \subseteq [n]$ by $\chi_S \in \{0, 1\}^n$, i.e., $\chi_i = 1$ if $i \in S$ and $\chi_i = 0$, otherwise.

We first state the intuitive result that after fixing $x_i = 1$ for $i \in N_1 \subseteq [n]$ and fixing $x_i = 0$ for $i \in N_0$ (with $N_0 \cap N_1 = \emptyset$), we again obtain a packing problem with a convex and quadratic packing constraint.

Lemma 1. *Let $W \in \mathbb{N}^{n \times n}$ be symmetric psd, $p \in \mathbb{N}^n$, and $c \in \mathbb{N}$. Further, let $N_0, N_1 \in 2^{[n]}$ with $N_0 \cap N_1 = \emptyset$ and $N_0 \cup N_1 \subsetneq [n]$ be arbitrary. Then, there exist $\tilde{n} \in \mathbb{N}$, $\tilde{W} \in \mathbb{N}^{\tilde{n} \times \tilde{n}}$ symmetric psd, $\tilde{p} \in \mathbb{N}^{\tilde{n}}$, and $\tilde{c} \in \mathbb{N}$ such that*

$$\begin{aligned} & \max \{p^\top x : x^\top W x \leq c, x \in \{0, 1\}^n, x_i = 0 \forall i \in N_0, x_i = 1 \forall i \in N_1\} \\ & = p^\top \chi_{N_1} + \max \{\tilde{p}^\top \tilde{x} : \tilde{x}^\top \tilde{W} \tilde{x} \leq \tilde{c}, \tilde{x} \in \{0, 1\}^{\tilde{n}}\}. \end{aligned}$$

Proof. Let $n_0 = |N_0|$, $n_1 = |N_1|$, and $\tilde{n} := n - n_0 - n_1$. Without loss of generality we can assume that $[\tilde{n}] = [n] \setminus (N_0 \cup N_1)$. Consider the matrix $\tilde{W} = (\tilde{w}_{ij}) \in \mathbb{N}^{\tilde{n} \times \tilde{n}}$ defined as

$$\tilde{w}_{ij} = \begin{cases} w_{ij} & \text{if } i \neq j, \\ w_{ij} + 2 \sum_{k \in N_1} w_{ik} & \text{if } i = j, \end{cases} \quad i, j \in [\tilde{n}].$$

Note that \tilde{W} is obtained from W by taking principal minors and adding diagonal matrices with non-negative entries so that \tilde{W} is positive semi-definite. Let $\tilde{c} = c - \chi_{N_1}^\top W \chi_{N_1}$. With a slight abuse of notation, for a set $S \subseteq [\tilde{n}]$, let $\tilde{\chi}_S$ denote its characteristic vector in $\{0, 1\}^{\tilde{n}}$ and χ_S its characteristic vector in $\{0, 1\}^n$. We then obtain for all $S \subseteq [\tilde{n}]$ the equality

$$\begin{aligned} \tilde{\chi}_S^\top \tilde{W} \tilde{\chi}_S &= \sum_{i \in S} \left(w_{ii} + 2 \sum_{k \in N_1} w_{ik} \right) + \sum_{i,j \in S: i \neq j} 2w_{ij} \\ &= \chi_{S \cup N_1}^\top W \chi_{S \cup N_1} - \chi_{N_1}^\top W \chi_{N_1}. \end{aligned}$$

Thus, we have $\tilde{\chi}^\top \tilde{W} \tilde{\chi}_S \leq \tilde{c}$ if and only if $\chi_{S \cup N_1}^\top W \chi_{S \cup N_1} \leq c$. Defining $\tilde{p} \in \mathbb{N}^{\tilde{n}}$ with $\tilde{p}_i = p_i$ for all $i \in [\tilde{n}]$ then establishes the claimed result. \square

By Lemma 1, the following assumptions are without loss of generality.

Lemma 2. *It is without loss of generality to assume that $0 < w_{ii} \leq c$ and $p_i > 0$ for all $i \in [n]$.*

Proof. If $w_{ii} > c$ for some $i \in [n]$, then $x_i = 0$ in every feasible solution x . If $w_{ii} = 0$, then the positive semi-definiteness of W implies $w_{ij} = w_{ji} = 0$ for every $j \in [n]$. Hence, the value of x_i does not influence the value of $x^\top W x$ and it is without loss of generality to assume that $x_i = 1$. Furthermore, if $p_i = 0$ then the value of x_i does not influence the value of $p^\top x$ and it is without loss of generality to assume that $x_i = 0$. In all cases, Lemma 1 yields the claimed result. \square

3 A Golden Ratio Approximation Algorithm

In this section, we derive a ϕ -approximation algorithm for packing problems with convex quadratic constraints of type (P) where $\phi = (\sqrt{5} - 1)/2 \approx 0.618$ is the inverse golden ratio. To this end, we first solve a convex relaxation of the problem. We then use the resulting solution to compute a feasible solution to another non-convex relaxation of the problem. The second relaxation has the property that any solution can be transformed so that it has at most one fractional value, and the transformation does not decrease the objective value. Together with a partial enumeration scheme in the spirit of Sahni [25], this yields a ϕ -approximation.

Denote by $d \in \mathbb{N}^n$ the diagonal of $W \in \mathbb{N}^{n \times n}$ and let $D := \text{diag}(d) \in \mathbb{N}^{n \times n}$ be the corresponding diagonal matrix. For a vector $x \in \{0, 1\}^n$ we have $x_i^2 = x_i$ for all $i \in [n]$ and, thus, we obtain

$$x^\top W x \geq x^\top D x = d^\top x \quad \text{for all } x \in \{0, 1\}^n.$$

We arrive at the following relaxation of (P):

$$\begin{aligned} & \text{maximize} && [p^\top x] \\ & \text{subject to} && x^\top W x \leq c, \\ & && d^\top x \leq c, \\ & && x \in [0, 1]^n. \end{aligned} \tag{R_1}$$

The following lemma shows that we can compute an exact optimal solution to (R₁) in polynomial time.

Lemma 3. *The relaxation (R₁) can be solved exactly in polynomial time.*

Proof. For every $x \in [0, 1]^n$, we have $[p^\top x] \in P := \{0, \dots, \sum_{i \in [n]} p_i\}$. For fixed $q \in P$, consider the mathematical program

$$\begin{aligned} & \text{minimize} && x^\top W x \\ & \text{subject to} && p^\top x \geq q, \\ & && d^\top x \leq c, \\ & && x \in [0, 1]^n \end{aligned} \tag{D_q}$$

with optimal value $c(q)$. Since (D_q) is quadratic and convex with linear constraints, it can be solved exactly in polynomial time, see Kozlov et al. [16]. If $c(q) > c$, we conclude that the maximal value of (R_1) is strictly smaller than q . If $c(q) \leq c$, the corresponding solution x solves (R_1) with an objective value of q . With binary search over P , we can compute the maximal value $q^* \in P$ such that (D_q) has a solution of at most c . The thus computed value q^* is the maximal objective of (R_1) and the corresponding optimal solution x of (D_q) is an optimal solution of (R_1) . \square

We proceed to propose a second relaxation of (P) . To this end, note that for every $x \in \{0, 1\}^n$ we have

$$x^\top W x = x^\top (W - D)x + x^\top D x = x^\top (W - D)x + d^\top x.$$

Relaxing the integrality condition yields the following relaxation of (P) :

$$\begin{aligned} & \text{maximize} && p^\top x \\ & \text{subject to} && x^\top (W - D)x + d^\top x \leq c, \\ & && x \in [0, 1]^n. \end{aligned} \tag{R_2}$$

Note that since the trace of $W - D$ is zero, $W - D$ has a negative eigenvalue unless all eigenvalues are zero. Hence, $W - D$ is not positive semi-definite, unless W is a diagonal matrix. Therefore, the relaxation (R_2) is in general not convex.

We proceed to show that (R_2) always has an optimal solution for which at most one variable is fractional. For $x \in \mathbb{R}^n$, let $N_0(x) := \{i \in [n] : x_i = 0\}$, $N_1(x) := \{i \in [n] : x_i = 1\}$, and $N_f(x) := [n] \setminus (N_1(x) \cup N_0(x))$.

Lemma 4. *For any feasible solution x of (R_2) , one can construct a feasible solution \bar{x} with $|N_f(\bar{x})| \leq 1$ and $p^\top \bar{x} \geq p^\top x$ in linear time.*

Proof. Let x be a feasible solution of (R_2) . Assume $|N_f(x)| \geq 2$, and consider $i, j \in N_f(x)$ with $i \neq j$, in particular, $x_i, x_j \in (0, 1)$. We proceed to construct a feasible solution \bar{x} with $|N_f(\bar{x})| \leq |N_f(x)| - 1$ and $p^\top \bar{x} \geq p^\top x$; see Figure 2 for an illustration.

Denote $v(x) := x^\top (W - D)x + d^\top x$, and for $k \in \{i, j\}$ let

$$\nu_k(x) := \frac{\partial}{\partial x_k} v(x) = \sum_{l \in [n] \setminus \{k\}} 2 w_{kl} x_l + w_{kk}, \quad r_k(x) := \frac{p_k}{\nu_k(x)}.$$

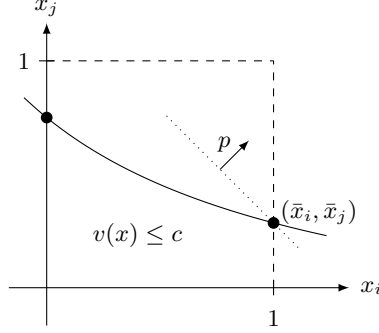


Fig. 2. Any feasible solution x of (R_2) with $|N_f(x)| \geq 2$ can be transformed into a feasible solution \bar{x} with $|N_f(\bar{x})| \leq |N_f(x)| - 1$ without decreasing the objective value.

By Lemma 2 it is without loss of generality to assume that $w_{kk} > 0$ and thus $\nu_k(x) > 0$. Note that $\nu_k(x)$ does not depend on x_k and therefore, for all $x \in \mathbb{R}^n$ and $t \in \mathbb{R}$, we have that

$$v(x + t\chi_k) = v(x) + t\nu_k(x), \quad (1)$$

where $\chi_k \in \{0, 1\}^n$ denotes the k -th unit vector.

Without loss of generality, assume that $r_i(x) \geq r_j(x)$ and define

$$\bar{\varepsilon} := \frac{\nu_i(x)}{\nu_j(x)}(1 - x_i), \quad \varepsilon := \min(x_j, \bar{\varepsilon}), \quad \delta := \frac{\nu_j(x)}{\nu_i(x)}\varepsilon.$$

Consider the vector $\bar{x} = x - \varepsilon\chi_j + \delta\chi_i$. By the definition of ε , we have $\bar{x}_j = x_j - \varepsilon \geq 0$. We further obtain

$$\bar{x}_i = x_i + \delta = x_i + \frac{\nu_j(x)}{\nu_i(x)}\varepsilon \leq x_i + \frac{\nu_j(x)}{\nu_i(x)}\bar{\varepsilon} = 1.$$

Note that $\bar{x}_j = 0$ if $\varepsilon = x_j$ and $\bar{x}_i = 1$ if $\varepsilon = \bar{\varepsilon}$ so that at least one of the inequalities $\bar{x}_j \geq 0$ and $\bar{x}_i \leq 1$ is tight. We conclude that $\bar{x} \in [0, 1]^n$ and $|N_f(\bar{x})| \leq |N_f(x)| - 1$. Furthermore, applying Equation (1), we get

$$\begin{aligned} v(\bar{x}) &= v(x - \varepsilon\chi_j + \delta\chi_i) \\ &= v(x - \varepsilon\chi_j) + \delta\nu_i(x - \varepsilon\chi_j) \\ &= v(x) - \varepsilon\nu_j(x) + \delta\nu_i(x - \varepsilon\chi_j) \\ &\leq v(x) - \varepsilon\nu_j(x) + \delta\nu_i(x) \\ &= v(x). \end{aligned}$$

Algorithm 1: Golden ratio algorithm

```

1 foreach  $H \subseteq [n]$  with  $|H| \leq 3$  do
2    $y^H \leftarrow$  sol. of  $(R_1)$  with  $x_i = 1 \forall i \in H, x_i = 0 \forall i \in \{j \in [n] \setminus H : p_j > \min_{h \in H} p_h\}$ 
3    $z^H \leftarrow$  transf. of  $\phi y^H$  containing at most one fractional variable
4    $\bar{z}^H \leftarrow \lfloor z^H \rfloor$ 
5  $H^* \leftarrow \operatorname{argmax} \{p^\top \bar{z}^H : H \subseteq [n] \text{ with } |H| \leq 3\}$ 
6 return  $\bar{z}^{H^*}$ 

```

Thus, \bar{x} is a feasible solution of (R_2) . Moreover, we have

$$\begin{aligned}
 p^\top \bar{x} &= p^\top x - \varepsilon p_j + \delta p_i \\
 &= p^\top x - \varepsilon p_j + \varepsilon \nu_j(x) \frac{p_i}{\nu_i(x)} \\
 &= p^\top x - \varepsilon p_j + \varepsilon \nu_j(x) r_i(x) \\
 &\geq p^\top x - \varepsilon p_j + \varepsilon \nu_j(x) r_j(x) \\
 &= p^\top x - \varepsilon p_j + \varepsilon p_j \\
 &= p^\top x.
 \end{aligned}$$

Applying this construction iteratively (at most) $|N_f(x)| - 1 \leq n - 1$ times yields the required result. \square

Remark 1. The algorithm in the proof of Lemma 4 can be improved by setting

$$\bar{\varepsilon} := \frac{(1 - x_i) \nu_i(x)}{\nu_j(x) + 2 w_{ij} (1 - x_i)}, \quad \varepsilon := \min(x_j, \bar{\varepsilon}), \quad \delta := \frac{\varepsilon \nu_j(x)}{\nu_i(x) - 2 w_{ij} x_j \varepsilon}.$$

In this way, we obtain $v(\bar{x}) = v(x)$ and increase the objective value at least as much as in the proof of Lemma 4 while still ensuring that \bar{x} is feasible for (R_2) and $|N_f(\bar{x})| \leq |N_f(x)| - 1$.

We proceed to devise a ϕ -approximation algorithm. The algorithm iterates over all sets $H \subseteq [n]$ with $|H| \leq 3$. For each set H it computes an optimal solution y^H to the convex relaxation (R_1) with the additional constraints

$$\begin{aligned}
 x_i &= 1 && \text{for all } i \in H, \text{ and} \\
 x_i &= 0 && \text{for all } i \in \{j \in [n] \setminus H : p_j > \min_{h \in H} p_h\}.
 \end{aligned}$$

Then, we scale down y^H by a factor of ϕ and show that ϕy^H is a feasible solution to the non-convex relaxation (R_2) . By Lemma 4, we can transform this solution into another solution z^H with at most one fractional variable. The integral part of z^H is our candidate solution for the starting set H . In the end, we return the best thus computed candidate over all possible sets H ; see Algorithm 1.

Theorem 1. *Algorithm 1 computes a ϕ -approximation for (P) .*

Proof. Fix an optimal solution x^* of (P) and define $S^* := \{i \in [n] : x_i^* = 1\}$. Since the algorithm iterates over all solutions of size at least three, it is without loss of generality for our following arguments to assume that $|S^*| \geq 4$. Let $H^* \subset S^*$ with $|H^*| = 3$ be chosen such that $p_i \leq \min_{h \in H^*} p_h$ for all $i \in S^*$ and consider the run of the algorithm when starting with H^* . Let $\bar{H} := \{i \in [n] \setminus H^* : p_i > \min_{h \in H^*} p_h\}$ and $k := |\bar{H}|$. It is without loss of generality to assume that $[n] \setminus (H^* \cup \bar{H}) = [n - k - 3]$. Consider the packing problem where as additional constraints we have $x_i = 1$ for all $i \in H^*$ and $x_i = 0$ for all $i \in \bar{H}$. By Lemma 1, this packing problem can be written as

$$\begin{aligned} & \text{maximize} && \tilde{p}^\top x \\ & \text{subject to} && x^\top \tilde{W} x \leq \tilde{c}, \\ & && x \in \{0, 1\}^{n-k-3}, \end{aligned} \tag{\tilde{P}}$$

where \tilde{W} is a symmetric and positive semi-definite matrix. We then have $p^\top x^* = \sum_{h \in H^*} p_h + \tilde{p}^\top \tilde{x}^*$ for an optimal solution \tilde{x}^* of (\tilde{P}) .

Let y be an optimal solution to the convex relaxation (R_1) of (\tilde{P}) . Since (R_1) is a relaxation of (\tilde{P}) , we have $p^\top y \geq p^\top \tilde{x}^*$. We proceed to show that ϕy is feasible for the non-convex relaxation (R_2) of (\tilde{P}) . To this end, we calculate

$$(\phi y)^\top (\tilde{W} - \tilde{D})(\phi y) + \phi \tilde{d}^\top y = \phi^2 y^\top (\tilde{W} - \tilde{D}) y + \phi \tilde{d}^\top y \leq \phi^2 \tilde{c} + \phi \tilde{c} = \tilde{c},$$

where for the inequality we used that y is feasible for the convex relaxation and, thus, $y^\top \tilde{W} y \leq \tilde{c}$ and $\tilde{d}^\top y \leq \tilde{c}$. By Lemma 4, we can transform ϕy into a solution z such that $\tilde{p}^\top z \geq \phi \tilde{p}^\top y$ and z has at most one fractional variable z_ℓ with $\ell \in [n - k - 3]$.

Let $S = H^* \cup \{i \in [n - k - 3] : z_i = 1\}$ and consider the solution χ_S . We have that

$$\chi_S^\top W \chi_S = \chi_{H^*}^\top W \chi_{H^*} + z^\top \tilde{W} z \leq \chi_{H^*}^\top W \chi_{H^*} + \tilde{c} = c,$$

so that χ_S is feasible for (P) . Moreover, we obtain

$$\begin{aligned} p^\top \chi_S &= \sum_{h \in H^*} p_h + \tilde{p}^\top z - p_\ell z_\ell \\ &\geq \sum_{h \in H^*} p_h + \phi \tilde{p}^\top y - p_\ell \\ &\geq \sum_{h \in H^*} p_h + \phi \tilde{p}^\top \tilde{x}^* - p_\ell \\ &= \phi p^\top x^* + (1 - \phi) \sum_{h \in H^*} p_h - p_\ell \\ &\geq \phi p^\top x^* + (3(1 - \phi) - 1) \min_{h \in H^*} p_h \\ &\geq \phi p^\top x^*, \end{aligned}$$

establishing the claimed result. \square

As a result of Theorem 1, we can derive an upper bound on the optimal value of (R_1) . This will turn out to be useful when constructing a monotone greedy algorithm in the next section.

Corollary 1. *Let x^* and y^* be optimal solutions to (P) and (R_1) , respectively. Then $p^\top y^* \leq \frac{2}{\phi} p^\top x^*$.*

Proof. Since y^* is feasible for (R_1) , we have

$$(\phi y^*)^\top (W - D)(\phi y^*) + d^\top (\phi y^*) \leq \phi^2 (y^*)^\top W y^* + \phi d^\top y^* \leq (\phi^2 + \phi)c = c.$$

Therefore, ϕy^* is feasible for (R_2) . By Lemma 4, we can transform ϕy^* into a vector z with $p^\top z \geq p^\top (\phi y^*) = \phi p^\top y^*$ and $|N_f(z)| \leq 1$. The integral part $\lfloor z \rfloor$ of z is feasible for (P) , and thus, $p^\top z \leq p^\top \lfloor z \rfloor + \max_{i \in [n]} p_i \leq 2p^\top x^*$. We conclude that $p^\top y^* \leq \frac{1}{\phi} p^\top z \leq \frac{2}{\phi} p^\top x^*$. \square

4 The Greedy Algorithm

In this section we analyze the greedy algorithm and show that, when combined with a partial enumeration scheme in the spirit of Sahni [25], it is at least a $(1 - \sqrt{3}/e)$ -approximation for packing problems with quadratic constraints of type (P) . We further show that its approximation ratio can be bounded from above by the golden ratio ϕ . Even though this approximation ratio is thus not better than the one guaranteed by the golden ratio algorithm (Theorem 1), it is worth analyzing it for several reasons. Firstly, it is simple to understand as well as to implement and turns out to have a much better running time in practice than the golden ratio algorithm; see the computational results in Section 8. And, secondly, the greedy algorithm serves as a main building block to devise a strategyproof mechanism with constant welfare guarantee; see Section 5.

For a set $S \subseteq [n]$, we write $w(S) := \chi_S^\top W \chi_S$. The core idea of the greedy algorithm is as follows. Assume that we have an initial solution $S \subset [n]$. Amongst all remaining items in $[n] \setminus S$, we pick an item i that maximizes the ratio between profit gain and weight gain, i.e.,

$$i \in \operatorname{argmax}_{j \in [n] \setminus S} \frac{p_j}{w(S \cup \{j\}) - w(S)}.$$

If adding i to the solution set would make it infeasible, i.e., $w(S \cup \{i\}) > c$, then we delete i from $[n]$. Otherwise, we add i to S . We repeat this process until $[n] \setminus S$ is empty.

It is known from the knapsack problem that, when starting the greedy algorithm as described above with the empty set as initial set, then the produced solution can be arbitrarily bad compared to an optimal solution. However, the greedy algorithm can be turned into a constant-factor approximation by using partial enumeration: For all feasible subsets $U \subseteq [n]$ with $|U| \leq 2$, we run the greedy algorithm starting with U as initial set. In the end we return the best solution set found in this process; see Algorithm 2.

Algorithm 2: Greedy algorithm with partial enumeration

```

1 foreach  $U \subseteq [n]$  with  $|U| \leq 2$  do
2    $S \leftarrow U, I \leftarrow [n]$ 
3   while  $I \setminus S \neq \emptyset$  do
4      $i \leftarrow \operatorname{argmax}_{j \in I \setminus S} \frac{p_j}{w(S \cup \{j\}) - w(S)}$ 
5     if  $w(S \cup \{i\}) \leq c$  then
6        $S \leftarrow S \cup \{i\}$ 
7     else
8        $I \leftarrow I \setminus \{i\}$ 
9    $S_U \leftarrow S$ 
10  $U^* \leftarrow \operatorname{argmax} \{p^\top \chi_{S_U} : U \subseteq [n] : |U| \leq 2\}$ 
11 return  $\chi_{S_{U^*}}$ 

```

The analysis of the algorithm follows a similar approach as the analysis of Sviridenko [27] for the greedy algorithm for maximizing a submodular function under a linear knapsack constraint. The non-linearity of the constraint in our case makes the analysis more complicated, though. In order to prove the approximation ratio of the greedy algorithm we need the following two technical lemmas.

Lemma 5. *Let $m \in \mathbb{N}$ and consider the sequence $(\theta_t)_{t \in \mathbb{N}}$ defined by the recursive formula*

$$1 - (m + 2\sqrt{tm})\theta_{t+1} = \sum_{i=1}^t \theta_i, \quad \theta_1 = \frac{1}{m}.$$

Then $\sum_{t=1}^m \theta_t \geq 1 - \frac{\sqrt{3}}{e}$.

Proof. Consider the initial value problem

$$\psi'(x) = \frac{1 - \psi(x)}{1 + 2\sqrt{x}}, \quad x \in [0, 1], \quad \psi(0) = 0.$$

Since the function $f: [0, 1] \times \mathbb{R} \rightarrow \mathbb{R}$, $(x, s) \mapsto \frac{1-s}{1+2\sqrt{x}}$ is Lipschitz-continuous in s , by the Picard-Lindelöf Theorem, this problem has a unique solution, which is given by

$$\psi(x) = 1 - e^{-\sqrt{x}} \sqrt{1 + 2\sqrt{x}}, \quad x \in [0, 1].$$

Since its first derivative

$$\psi'(x) = \frac{e^{-\sqrt{x}}}{\sqrt{1 + 2\sqrt{x}}}, \quad x \in [0, 1],$$

is monotonically decreasing, it follows that ψ is concave.

Define $z_t := \sum_{i=1}^t \theta_i$, $t \in \{0, \dots, m\}$. We claim that for every $t \in \{0, \dots, m\}$ we have

$$z_t \geq \psi\left(\frac{t}{m}\right). \tag{2}$$

Note that (2) implies the result using

$$\sum_{t=1}^m \theta_t = z_m \geq \psi(1) = 1 - \frac{\sqrt{3}}{e}.$$

To finish the proof, we prove (2) by induction. We have $z_0 = 0 = \psi(0)$. Now assume that (2) holds for some arbitrary but fixed $t \in \{0, \dots, m-1\}$. By the recursive definition of $(\theta_t)_{t \in \mathbb{N}}$ and the concavity of ψ , it then follows that

$$\begin{aligned} z_{t+1} &= z_t + \theta_{t+1} \\ &= z_t + \frac{1 - z_t}{m + 2\sqrt{tm}} \\ &= z_t \left(1 - \frac{1}{m + 2\sqrt{tm}}\right) + \frac{1}{m + 2\sqrt{tm}} \\ &\geq \psi\left(\frac{t}{m}\right) \left(1 - \frac{1}{m + 2\sqrt{tm}}\right) + \frac{1}{m + 2\sqrt{tm}} \\ &= \psi\left(\frac{t}{m}\right) + \frac{1 - \psi\left(\frac{t}{m}\right)}{m + 2\sqrt{tm}} \\ &= \psi\left(\frac{t}{m}\right) + \frac{1}{m} \psi'\left(\frac{t}{m}\right) \\ &\geq \psi\left(\frac{t+1}{m}\right), \end{aligned}$$

which completes the proof. \square

Lemma 6. Let $w_0, \dots, w_m \in \mathbb{N}$ with $0 = w_0 < w_1 < \dots < w_m$, and let $\theta_i \geq 0$, $i \in [m]$. Then,

$$\sum_{i=1}^m \theta_i (w_i - w_{i-1}) \geq \left(1 - \frac{\sqrt{3}}{e}\right) \min_{t=0, \dots, m-1} \sum_{i=1}^t \theta_i (w_i - w_{i-1}) + \theta_{t+1} (w_m + 2\sqrt{w_t w_m}).$$

Proof. We first show the statement for sequences $0 = w_0 < w_1 < \dots < w_m$ with the additional property that $w_i - w_{i-1} = 1$ for all $i \in [m]$. For this case, it is to show that

$$\sum_{i=1}^m \theta_i \geq \left(1 - \frac{\sqrt{3}}{e}\right) \min_{t=0, \dots, m-1} \sum_{i=1}^t \theta_i + \theta_{t+1} (m + 2\sqrt{tm}).$$

It suffices to show that the optimal value of the following optimization problem is at least $1 - \sqrt{3}/e$.

$$\begin{aligned} &\text{minimize} && \sum_{i=1}^m \theta_i \\ &\text{subject to} && \sum_{i=1}^t \theta_i + \theta_{t+1} (m + 2\sqrt{tm}) \geq 1 \quad \text{for all } t = 0, \dots, m-1, \\ &&& \theta_i \geq 0, \quad \text{for all } i = 1, \dots, m. \end{aligned} \tag{3}$$

We claim that every optimal solution to (3) satisfies all inequalities with equality. For a contradiction, fix an optimal solution $\theta_1^*, \dots, \theta_m^*$ and suppose there is $s \in \{0, \dots, m-1\}$ such that

$$\sum_{i=1}^s \theta_i^* + \theta_{s+1}^*(m + 2\sqrt{sm}) > 1.$$

Choosing the minimal s with this property, we have $\theta_{s+1}^* > 0$. Let

$$\delta = \min \left\{ \theta_{s+1}^*, \frac{\sum_{i=1}^s \theta_i^* + \theta_{s+1}^*(m + 2\sqrt{sm}) - 1}{m + 2\sqrt{sm}} \right\},$$

and consider the solution $\theta'_1, \dots, \theta'_m$ defined as

$$\theta'_i = \begin{cases} \theta_i^* & \text{if } i < s+1, \\ \theta_i^* - \delta & \text{if } i = s+1, \\ \theta_i^* + \frac{\delta}{m+2\sqrt{(i-1)m}} & \text{if } i > s+1. \end{cases}$$

We first check that the solution $\theta'_1, \dots, \theta'_m$ is feasible. For the inequalities for $t = 0, \dots, s-1$, there is nothing to show since the involved variables are not altered. For $t = s$, the inequality is satisfied by the choice of δ . For $t > s$, we obtain

$$\sum_{i=1}^t \theta'_i + \theta'_{t+1}(m + 2\sqrt{tm}) \geq \sum_{i=1}^t \theta_i^* - \delta + \left(\theta_{t+1}^* + \frac{\delta}{m + 2\sqrt{tm}} \right) (m + 2\sqrt{tm}) \geq 1,$$

where for the second inequality we used that $\theta_1^*, \dots, \theta_m^*$ is feasible. Finally, we note that

$$\sum_{i=1}^m \theta_i^* - \sum_{i=1}^m \theta'_i = \delta - \sum_{i=s+2}^m \frac{\delta}{m + 2\sqrt{(i-1)m}} \geq \delta \left(1 - \frac{m-1}{m} \right) > 0,$$

contradicting the optimality of $\theta_1^*, \dots, \theta_m^*$. We conclude that every optimal solution of (3) satisfies all inequalities with equality. The result then follows from Lemma 5.

It is left to show that the statement holds for arbitrary finite sequences $0 = w_0 < w_1 < \dots < w_m$. Fix such a sequence, let $m' := w_m$, and let $\theta'_1, \dots, \theta_{m'}$ be such that there are first $w_1 - w_0$ copies of θ_1 , then $w_2 - w_1$ copies of θ_2 , and

so on. We thus obtain

$$\begin{aligned}
\sum_{i=1}^m \theta_i (w_i - w_{i-1}) &= \sum_{i=1}^{m'} \theta'_i \\
&\geq \left(1 - \frac{\sqrt{3}}{e}\right) \min_{t=0, \dots, m'-1} \sum_{i=1}^t \theta'_i + \theta'_{t+1} (m' + 2\sqrt{tm'}) \\
&= \left(1 - \frac{\sqrt{3}}{e}\right) \min_{t=w_0, \dots, w_{m-1}} \sum_{i=1}^t \theta'_i + \theta'_{t+1} (m' + 2\sqrt{tm'}) \\
&= \left(1 - \frac{\sqrt{3}}{e}\right) \min_{t=0, \dots, m-1} \sum_{i=1}^t (w_i - w_{i-1}) \theta_i + \theta_{t+1} (w_m + 2\sqrt{tw_m}),
\end{aligned}$$

yielding the result. \square

We can now prove the approximation ratio of the greedy algorithm.

Theorem 2. *The Greedy algorithm with partial enumeration (Algorithm 2) is an approximation algorithm with approximation ratio $(1 - \frac{\sqrt{3}}{e})$ for (P).*

Proof. Let x^* be an optimal solution of (P) and set $S^* := \{i \in [n] : x_i^* = 1\}$. Number the items of $S^* = \{i_1^*, i_2^*, \dots, i_k^*\}$ such that $p_{i_1^*} \geq p_{i_2^*} \geq \dots \geq p_{i_k^*}$. Since the algorithm enumerates all solutions with at most two items, it is without loss of generality to assume that $|S^*| \geq 3$. Consider the run of the greedy algorithm with $U = \{i_1^*, i_2^*\}$. Without loss of generality, we assume that $i_1^* = n - 1$ and $i_2^* = n$. Set $S^0 := U$, and for $t = 1, 2, \dots$, denote by S^t and i_t the values of S and i after the t -th pass of the **while** loop. Furthermore, define

$$\theta_t := \frac{p_{i_t}}{w(S^{t-1} \cup \{i_t\}) - w(S^{t-1})}.$$

By Lemma 1, we can treat the problem after fixing $x_{i_1^*} = x_{i_2^*} = 1$ as a new problem of the same form with matrix $\tilde{W} \in \mathbb{N}^{(n-2) \times (n-2)}$, profit vector \tilde{p} , and budget \tilde{c} . In the following, for a set $S \subseteq [n] \setminus U$ we write $\tilde{w}(S) := \chi_S^\top \tilde{W} \chi_S$. Note that \tilde{w} is supermodular, i.e., for any two sets $S, S' \subseteq [n] \setminus U$ we have

$$\sum_{i \in S' \setminus S} \tilde{w}(S \cup \{i\}) - \tilde{w}(S) \leq \tilde{w}(S \cup S') - \tilde{w}(S).$$

By Lemma 2, we can assume without loss of generality that $\tilde{w}(S \cup \{i\}) - \tilde{w}(S) > 0$.

Let t^* be the first step of the greedy algorithm for which $i_{t^*} \in S^*$ but the algorithm does not add i_{t^*} to its solution set. It is without loss of generality to assume that in all previous iterations $t \in \{1, \dots, t^* - 1\}$ we had $S^t = S^{t-1} \cup \{i_t\}$ as otherwise item i_t would be neither contained in the optimal solution nor the solution computed by the greedy algorithm; thus, removing it from the instance would not change the analysis. Since i_{t^*} is not included in the solution, we

have $\tilde{w}(S^{t^*-1} \cup \{i_{t^*}\} \setminus U) > \tilde{c}$. In the following, we write $S^{t^*} := S^{t^*-1} \cup \{i_{t^*}\}$, $\tilde{S}^* := S^* \setminus U$, and for $t \in \{0, \dots, t^*\}$, we write $\tilde{S}^t := S^t \setminus U$.

For all $t \in \{0, \dots, t^* - 1\}$, we obtain

$$\begin{aligned} \sum_{i \in \tilde{S}^*} p_i &\leq \sum_{i \in \tilde{S}^t} p_i + \sum_{i \in \tilde{S}^* \setminus \tilde{S}^t} p_i \\ &= \sum_{i \in \tilde{S}^t} p_i + \sum_{i \in \tilde{S}^* \setminus \tilde{S}^t} \frac{p_i}{\tilde{w}(\tilde{S}^t \cup \{i\}) - \tilde{w}(\tilde{S}^t)} (\tilde{w}(\tilde{S}^t \cup \{i\}) - \tilde{w}(\tilde{S}^t)) \\ &\leq \sum_{i \in \tilde{S}^t} p_i + \theta_{t+1} \sum_{i \in \tilde{S}^* \setminus \tilde{S}^t} (\tilde{w}(\tilde{S}^t \cup \{i\}) - \tilde{w}(\tilde{S}^t)) \\ &\leq \sum_{i \in \tilde{S}^t} p_i + \theta_{t+1} (\tilde{w}(\tilde{S}^t \cup \tilde{S}^*) - \tilde{w}(\tilde{S}^t)), \end{aligned}$$

where we used the supermodularity of \tilde{w} . By the Cauchy-Schwarz inequality it holds that

$$\begin{aligned} \tilde{w}(\tilde{S}^t \cup \tilde{S}^*) - \tilde{w}(\tilde{S}^t) &= (\chi_{\tilde{S}^t} + \chi_{\tilde{S}^* \setminus \tilde{S}^t})^\top \tilde{W} (\chi_{\tilde{S}^t} + \chi_{\tilde{S}^* \setminus \tilde{S}^t}) - w(\tilde{S}^t) \\ &\leq \tilde{w}(\tilde{S}^* \setminus \tilde{S}^t) + 2\sqrt{\tilde{w}(\tilde{S}^t) \tilde{w}(\tilde{S}^* \setminus \tilde{S}^t)} \\ &\leq \tilde{c} + 2\sqrt{\tilde{w}(\tilde{S}^t) \tilde{c}}. \end{aligned}$$

Thus, we get

$$\sum_{i \in \tilde{S}^*} p_i \leq \sum_{i \in \tilde{S}^t} p_i + \theta_{t+1} (\tilde{c} + 2\sqrt{\tilde{w}(\tilde{S}^t) \tilde{c}}) \quad \text{for all } t \in \{0, \dots, t^* - 1\}.$$

Since $\tilde{c} < \tilde{w}(\tilde{S}^{t^*})$, it follows that

$$\begin{aligned} \sum_{i \in \tilde{S}^*} p_i &\leq \min_{t=0, \dots, t^*-1} \sum_{i \in \tilde{S}^t} p_i + \theta_{t+1} (\tilde{w}(\tilde{S}^{t^*}) + 2\sqrt{\tilde{w}(\tilde{S}^{t^*}) \tilde{w}(\tilde{S}^t)}) \\ &= \min_{t=0, \dots, t^*} \sum_{i=1}^t \theta_i (\tilde{w}(\tilde{S}^i) - \tilde{w}(\tilde{S}^{i-1})) + \theta_{t+1} (\tilde{w}(\tilde{S}^{t^*}) + 2\sqrt{\tilde{w}(\tilde{S}^{t^*}) \tilde{w}(\tilde{S}^t)}). \end{aligned}$$

Furthermore,

$$\sum_{i \in \tilde{S}^{t^*}} p_i = \sum_{i=1}^{t^*} \theta_i (\tilde{w}(\tilde{S}^i) - \tilde{w}(\tilde{S}^{i-1}))$$

and thus, by Lemma 6,

$$\sum_{i \in \tilde{S}^{t^*}} p_i \geq \left(1 - \frac{\sqrt{3}}{e}\right) \sum_{i \in \tilde{S}^*} p_i.$$

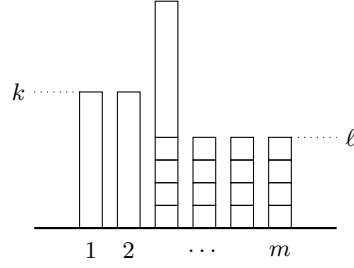


Fig. 3. A partial greedy solution $S = (S_1, S_2)$ with initial set $S^0 = (S_1^0, S_2^0)$, where $S_1^0 = \{1, 2\}$ and $S_2^0 = \emptyset$. The long bars represent type 1 items whereas the short bars represent type 2 items.

Finally, this leads to

$$\begin{aligned}
\sum_{i \in S^{t^*-1}} p_i &= \sum_{i \in U} p_i + \sum_{i \in \tilde{S}^{t^*-1}} p_i \\
&= \sum_{i \in U} p_i + \sum_{i \in \tilde{S}^{t^*}} p_i - p_{i_{t^*}} \\
&\geq \sum_{i \in U} p_i + \left(1 - \frac{\sqrt{3}}{e}\right) \sum_{i \in \tilde{S}^*} p_i - p_{i_{t^*}} \\
&\geq \sum_{i \in U} p_i + \left(1 - \frac{\sqrt{3}}{e}\right) \sum_{i \in \tilde{S}^*} p_i - \frac{1}{2} \sum_{i \in U} p_i \\
&\geq \left(1 - \frac{\sqrt{3}}{e}\right) \sum_{i \in S^*} p_i.
\end{aligned}$$

Since the greedy algorithm with starting solution U obtains a profit of at least $\sum_{i \in S^{t^*-1}} p_i$, this implies the claimed result. \square

We proceed to show that the approximation ratio of the greedy algorithm can be bounded from above by the golden ratio.

Theorem 3. *The approximation ratio of the greedy algorithm with partial enumeration is at most $\phi = (\sqrt{5} - 1)/2$, even if we allow partial enumeration over an arbitrary but fixed number of items.*

Proof. Consider the following instance. Let $m, \ell, k \in \mathbb{N}$ with $\ell < k$ and denote by χ_i the i -th unit vector in \mathbb{R}^m . Let there be two types of items: m items of type 1 with profit $p^{(1)} = 1$ and weight vector $y_i^{(1)} = k\chi_i$, $i \in [m]$, and $m\ell$ type 2 items with profit $p^{(2)} = \frac{1+2\ell}{k^2+2k\ell}$ and weight vector $y_i^{(2)} = \chi_{\lceil \frac{i}{\ell} \rceil}$, $i \in [m\ell]$; see Figure 3 for an illustration.

We wish to solve

$$\begin{aligned}
& \text{maximize} && p(S_1, S_2) := |S_1|p^{(1)} + |S_2|p^{(2)} \\
& \text{subject to} && w(S_1, S_2) := \left\| \sum_{i \in S_1} y_i^{(1)} + \sum_{i \in S_2} y_i^{(2)} \right\|_2^2 \leq mk^2, \\
& && S_1 \subseteq [m], S_2 \subseteq [m\ell].
\end{aligned}$$

Setting $Y := [y_1^{(1)}, \dots, y_m^{(1)}, y_1^{(2)}, \dots, y_{m\ell}^{(2)}]$, this optimization problem can be reformulated as in (P) with weight matrix $W = Y^\top Y$, which is clearly nonnegative and positive semidefinite.

We first derive the solution produced by the greedy algorithm. Partition $[m\ell] = \bigcup_{i=1}^m T_i$, where $T_i = \{j \in [m\ell] : \lceil \frac{j}{\ell} \rceil = i\}$. Consider a partial greedy solution $S = (S_1, S_2)$ and assume that $i \notin S_1$ and $j \notin S_2$ for some type 1 item $i \in [m]$ and type 2 item $j \in T_i$. Let $h := |S_2 \cap T_i| < \ell$. Then we have $w(S_1, S_2 \cup \{j\}) - w(S_1, S_2) = (h+1)^2 - h^2 = 1 + 2h$, and thus

$$\begin{aligned}
\frac{p^{(2)}}{w(S_1, S_2 \cup \{j\}) - w(S_1, S_2)} &= \frac{1 + 2\ell}{(k^2 + 2k\ell)(1 + 2h)} \\
&> \frac{1 + 2\ell}{(k^2 + 2kh)(1 + 2\ell)} \\
&= \frac{1}{k^2 + 2kh} \\
&= \frac{p^{(1)}}{w(S_1 \cup \{i\}, S_2) - w(S_1, S_2)}.
\end{aligned}$$

Hence, the greedy algorithm will always include type 2 item $j \in T_i$ before type 1 item i in its solution.

Assume that for a partial solution $S = (S_1, S_2)$ we have $i \in S_1$, $i' \notin S_1$, and $j \notin S_2$ for some type 1 items $i, i' \in [m]$ and a type 2 item $j \in T_i$. Since $|S_2 \cap T_{i'}| \leq \ell$, we have

$$\begin{aligned}
\frac{p^{(1)}}{w(S_1 \cup \{i'\}, S_2) - w(S_1, S_2)} &\geq \frac{1}{k^2 + 2k\ell} \\
&> \frac{1 + 2\ell}{(1 + 2k)(k^2 + 2k\ell)} \\
&\geq \frac{p^{(2)}}{w(S_1, S_2 \cup \{j\}) - w(S_1, S_2)}.
\end{aligned}$$

Consequently, the greedy algorithm will always add type 1 item i' before type 2 item $j \in T_i$ to its solution given that type 1 item i is already included.

Thus, the greedy algorithm starts with some initial solution $S^0 = (S_1^0, S_2^0)$. Afterwards, it includes all type 2 items in $[m\ell] \setminus \bigcup_{i \in S_1^0} T_i$ (Step 1). Finally, it adds type 1 items until the capacity bound of mk^2 is reached (Step 2). Let $s := |S_1^0|$. The weight of the partial solution after Step 1 is given by $sk^2 + (m-s)\ell^2$. Adding

any type 1 item in Step 2 increases the weight of the solution by $k^2 + 2k\ell$. Hence, in Step 2,

$$r := \frac{mk^2 - sk^2 - (m-s)\ell^2}{k^2 + 2k\ell} = \frac{(m-s)(k^2 - \ell^2)}{k^2 + 2k\ell}$$

type 1 items are added until the capacity is reached. (It is without loss of generality to assume that $r \in \mathbb{Z}$ since otherwise after adding $\lfloor r \rfloor$ type 1 items, the remaining capacity would be filled with type 2 items and the resulting approximation ratio would be even lower.) Thus, the profit of the solution \hat{S} produced by the greedy algorithm is given by

$$\begin{aligned} p(\hat{S}) &= (s+r)p^{(1)} + (m-s)\ell p^{(2)} \\ &= s + \frac{(m-s)(k^2 - \ell^2)}{k^2 + 2k\ell} + \frac{(m-s)\ell(1+2\ell)}{k^2 + 2k\ell} \\ &= s + \frac{(m-s)(k^2 + \ell^2 + \ell)}{k^2 + 2k\ell} \\ &= m \left[\frac{s}{m} + \frac{(1 - \frac{s}{m})(1 + \frac{\ell^2}{k^2} + \frac{\ell}{k})}{1 + 2\frac{\ell}{k}} \right] \\ &= m \left[\frac{s}{m} + \frac{(1 - \frac{s}{m})(1 + q^2 + \frac{q}{k})}{1 + 2q} \right], \end{aligned}$$

where $q := \frac{\ell}{k}$.

On the other hand, consider the solution $S^* = (S_1^*, S_2^*)$ with $S_1^* = [m]$ and $S_2^* = \emptyset$. It fulfills $p(S) = m$ and $w(S) = mk^2$. Thus, we have

$$\rho(q) := \lim_{k,m \rightarrow \infty} \frac{p(\hat{S})}{p(S^*)} = \frac{1 + q^2}{1 + 2q}.$$

Under the constraint $q \in (0, 1)$, the ratio $\rho(q)$ attains its minimum at $q = \phi$ with value $\rho(\phi) = \phi$, where $\phi = \frac{\sqrt{5}-1}{2}$ is the golden ratio. \square

5 Monotone Algorithms

To illustrate the need for monotone algorithms, reconsider the situation described in Example 1 with a set of n selfish agents requesting permission to send gas through a pipeline. Each agent j has a private value p_j expressing the monetary gain from being allowed to send the gas. A natural objective of a system provider is to maximize social welfare, i.e., to solve (P). Since the true value p_j is the private information of agent j , the system designer has to employ a mechanism that incentivizes the agents to report their true values p_j .² It is without loss of generality [9,19] to assume the following form of a direct revelation

² We here make the standard assumption that the true values of the source vertex s_j , the target vertex t_j , and the quantity of gas q_j are public knowledge. This is reasonable since these values are physically measurable by the system provider so that misreporting them would be pointless for the agent. This assumption is also frequently made in the knapsack auction literature [1,4,18].

Algorithm 3: Monotone greedy algorithm

```
1  $y^* \leftarrow$  solution of  $(R_1)$ 
2 if  $\max_{i \in [n]} p_i \geq (1 - \frac{\sqrt{3}}{e}) / (1 + \frac{4}{\sqrt{5}-1}) p^\top y^*$  then
3   return  $\chi_{i^*}$  for  $i^* \in \operatorname{argmax}_{i \in [n]} p_i$ 
4 else
5   return solution of Greedy algorithm without partial enumeration.
```

mechanism. The mechanism elicits a (potentially misrepresented) bid p'_j from each agent j and computes a solution $x(p') \in \{0, 1\}^n$ to (P) based on these values. Further, the mechanism computes a payment g_j for each agent j . The utility of agent j , when their true valuation is p_j and the agents report p' , is then $p_j x_j(p') - g_j(p')$. The mechanism is strategyproof if truth-telling is a dominant strategy of each agent j in the underlying game where each agent chooses a value to report.

Myerson [19] shows that an algorithm \mathcal{A} can be turned into a strategyproof mechanism if and only if it is monotone in the following sense. Let $x(p')$ denote the feasible solution to (P) computed by \mathcal{A} as a function of the reported valuations. Then \mathcal{A} is monotone if for all agents j the function $x_j(p')$ is nondecreasing in p'_j for all fixed values p'_i with $i \neq j$. For a monotone algorithm, charging every agent j with $x_j(p') = 1$ the critical bid $\inf \{z \in \mathbb{R}_{\geq 0} : x_j(z, p'_{-j}) = 1\}$ and charging all other agents nothing yields a strategyproof mechanism. Here, $x_j(z, p'_{-j})$ denotes the binary variable x_j as a function of the bid of agent j , when the bids p'_{-j} of the other agents are fixed.

We note that the algorithms designed in Sections 3 and 4 are unlikely to be monotone, since the partial enumeration schemes in both of them are not monotone. On the other hand, without the enumeration scheme, they do not provide a constant approximation, even when W is a diagonal matrix. However, by combining ideas from both algorithm, we derive a monotone algorithm with constant approximation guarantee.

Theorem 4. *Algorithm 3 is a monotone α -approximation algorithm for (P) , where $\alpha = (1 - \frac{\sqrt{3}}{e}) / (1 + \frac{4}{\sqrt{5}-1}) \approx 0.086$. The corresponding critical payments can be computed in polynomial time.*

Proof. We first prove the approximation ratio. For $\phi := \frac{\sqrt{5}-1}{2}$ and $\rho := 1 - \frac{\sqrt{3}}{e}$, we have $\alpha = \rho / (1 + \frac{2}{\phi})$. Let p^* and q^* be the optimal values of Problems (P) and (R_1) , respectively. Since (R_1) is a relaxation of (P) we have that $q^* \geq p^*$. If $p_i \geq \alpha q^*$ for some $i \in [n]$ it follows that $p^\top \chi_i \geq \alpha p^*$.

Assume that $p_i < \alpha q^*$ for all $i \in [n]$, and let x be the solution computed by the greedy algorithm (without partial enumeration). Following the proof of Theorem 2 we see that $p^\top x \geq \rho p^* - p_j$ for some item $j \in [n]$. Since $p_j < \alpha q^*$, and by Corollary 1 we have $q^* \leq \frac{2}{\phi} p^*$, we obtain

$$p^\top x \geq \left(\rho - \frac{2\alpha}{\phi} \right) p^* = \alpha p^*.$$

Next, we prove the monotonicity of the algorithm. To this end, let $p, \hat{p} \in \mathbb{N}^n$ be two declared profit vectors such that there is $i \in [n]$ with $\hat{p}_i = p_i + 1$ and $\hat{p}_j = p_j$ for all $j \neq i$. Let x and \hat{x} be the corresponding solutions computed by Algorithm 3 and assume that $x_i = 1$. It is to show that $\hat{x}_i = 1$. Let q^* and \hat{q}^* be the optimal values of (R_1) with respect to p and \hat{p} . Then $q^* \leq \hat{q}^* \leq q^* + 1$. Let $H := \{j \in [n] : p_j \geq \alpha q^*\}$ and $\hat{H} := \{j \in [n] : \hat{p}_j \geq \alpha \hat{q}^*\}$.

First, assume that $H \neq \emptyset$. Since by assumption $x_i = 1$, it follows that $p_i \geq \alpha q^*$ and $p_i = \max_{j \in [n]} p_j$. Therefore,

$$\hat{p}_i = p_i + 1 \geq \alpha q^* + 1 \geq \alpha(q^* + 1) \geq \alpha \hat{q}^*,$$

and thus $\hat{H} \neq \emptyset$. Furthermore, i is the only item in $\operatorname{argmax}_{j \in [n]} \hat{p}_j$ and hence $\hat{x}_i = 1$.

Next, assume that $H = \emptyset$. Then either $\hat{H} = \{i\}$ and thus $\hat{x}_i = 1$, or $\hat{H} = \emptyset$. In the latter case, Algorithm 3 executes the greedy algorithm for both p and \hat{p} . But since for every $S \subseteq [n] \setminus \{i\}$

$$\frac{\hat{p}_i}{w(S \cup \{i\}) - w(S)} > \frac{p_i}{w(S \cup \{i\}) - w(S)}$$

and for every $j \in [n] \setminus \{i\}$ and every $S \subseteq [n] \setminus \{j\}$

$$\frac{\hat{p}_j}{w(S \cup \{j\}) - w(S)} = \frac{p_j}{w(S \cup \{j\}) - w(S)},$$

when the greedy algorithm adds item i to its solution after k iterations for p , then it also adds i to its solution after at most k iterations for \hat{p} .

The critical payments can be computed with binary search. \square

6 Constantly Many Packing Constraints

In this section we generalize Problem (P) by allowing a constant number of convex quadratic constraints and derive a constant-factor approximation algorithm using randomized rounding combined with partial enumeration. To this end, let $r \in \mathbb{N}$ be a constant natural number, and for every $k \in [r]$ let $W^k = (w_{ij}^k)_{i,j \in [n]} \in \mathbb{N}^{n \times n}$ be a symmetric psd matrix with non-negative entries. Furthermore, let $p \in \mathbb{N}^n$ and $c^k \in \mathbb{N}$, $k \in [r]$. We consider packing problems with r convex quadratic knapsack constraints of the form

$$\begin{aligned} & \text{maximize} && p^\top x \\ & \text{subject to} && x^\top W^k x \leq c^k \quad \text{for all } k \in [r], \\ & && x \in \{0, 1\}^n. \end{aligned} \tag{P^k}$$

Algorithm 4: Randomized rounding

```

1  $y \leftarrow \varepsilon$ -optimal solution of  $(R^k)$ 
2 repeat
3    $x \leftarrow$  realization of  $\text{Ber}(\alpha y)$ 
4 until  $x$  is feasible for  $(P^k)$ 
5 return  $x$ 

```

Denote by d^k the vector consisting of the diagonal elements of W^k . We obtain the following convex relaxation of (P^k) ,

$$\begin{aligned}
 & \text{maximize} && p^\top x \\
 & \text{subject to} && x^\top W^k x \leq c^k \quad \text{for all } k \in [r], \\
 & && (d^k)^\top x \leq c^k \quad \text{for all } k \in [r], \\
 & && x \in [0, 1]^n.
 \end{aligned} \tag{R^k}$$

For $\varepsilon > 0$ we call a solution y of (R^k) ε -optimal if $p^\top y \geq (1 - \varepsilon)q^*$, where q^* is the optimal value of (R^k) . Convex problems of type (R^k) can be solved ε -optimally in polynomial time by interior points methods [20].

Lemma 7. *For every $\varepsilon > 0$ the relaxation (R^k) can be solved ε -optimally in polynomial time.*

We proceed to derive an approximation algorithm based on solving (R^k) . For some fixed value $\delta \in (0, 1)$, we call items i with $w_{ii}^k \leq \delta c^k$ for all $k \in [r]$ δ -light. All other items are called δ -heavy. We first assume that all items are δ -light and devise a randomized constant-factor approximation algorithm for (P^k) based on randomized rounding; see Algorithm 4. To that end, for some vector $y \in [0, 1]^n$, denote by $\text{Ber}(y)$ the vector of stochastically independent binary random variables $X = (X_1, \dots, X_n)^\top$ with the property $\mathbb{P}[X_i = 1] = y_i$ and $\mathbb{P}[X_i = 0] = 1 - y_i$, for $i \in [n]$.

Lemma 8. *Let $\delta \in (0, 1)$ and assume that all items $i \in [n]$ are δ -light. Let $\varepsilon \in (0, 1)$, p^* be the optimal value of (P^k) , y be an ε -optimal solution of (R^k) , $\alpha \in (0, 1)$, and $X = \text{Ber}(\alpha y)$. Then, $\mathbb{E}[p^\top X \mid X \text{ is feasible}] \geq f(\alpha, \delta)(1 - \varepsilon)p^*$, where $f(\alpha, \delta) = \alpha(1 - g(\alpha, \delta))^\tau$ and $g(\alpha, \delta) = \alpha(1 + (1 + \delta^{\frac{1}{3}})^3) + (1 - \alpha)\delta$.*

Proof. By Bayes' theorem we have

$$\begin{aligned}
 \mathbb{E}[p^\top X \mid X \text{ feasible}] &= \sum_{\ell \in [n]} p_\ell \mathbb{E}[X_\ell \mid X \text{ feasible}] \\
 &= \sum_{\ell \in [n]} p_\ell \mathbb{P}[X_\ell = 1 \mid X \text{ feasible}] \\
 &= \sum_{\ell \in [n]} p_\ell \frac{\mathbb{P}[X_\ell = 1] \mathbb{P}[X \text{ feasible} \mid X_\ell = 1]}{\mathbb{P}[X \text{ feasible}]}
 \end{aligned}$$

$$\begin{aligned}
&\geq \alpha \sum_{\ell \in [n]} p_\ell y_\ell \mathbb{P}[X^\top W^k X \leq c^k \text{ for all } k \in [r] \mid X_\ell = 1] \\
&\geq \alpha \sum_{\ell \in [n]} p_\ell y_\ell \prod_{k=1}^r \mathbb{P}[X^\top W^k X \leq c^k \mid X_\ell = 1], \tag{4}
\end{aligned}$$

where the last inequality follows from the monotonicity of the functions $X \mapsto X^\top W^k X$ with respect to the natural partial order on \mathbb{R}^n and the FKG inequality; see [7]. In the following, we show that for every $\ell \in [n]$ and $k \in [r]$ we have

$$\mathbb{P}[X^\top W^k X \leq c^k \mid X_\ell = 1] \geq 1 - g(\alpha, \delta). \tag{5}$$

Combining (4) and (5), and using that $p^\top y \geq (1 - \varepsilon)p^*$ we then obtain

$$\mathbb{E}[p^\top X \mid X \text{ feasible}] \geq \alpha p^\top y (1 - g(\alpha, \delta))^r = f(\alpha, \delta) p^\top y \geq f(\alpha, \delta)(1 - \varepsilon)p^*,$$

and we are finished.

We proceed as follows. Let $\ell \in [n]$ and $k \in [r]$. Define $z \in \mathbb{R}^n$ by $z_i = y_i$ for all $i \in [n] \setminus \{\ell\}$ and $z_\ell = 1$. We derive an upper bound on $z^\top W^k z$ and then use this bound to prove that $\mathbb{E}[X^\top W^k X \mid X_\ell = 1] \leq g(\alpha, \delta) c^k$. Using Markov's inequality yields (5). For the sake of readability, throughout the rest of the proof we omit the superscripts and simply write $W = W^k$ and $c = c^k$.

Claim. We have

$$z^\top W z \leq \min_{\gamma \in (0,1)} \left[\frac{\delta}{(1-\gamma)^2} + \frac{1}{\gamma^2} \right] c = (1 + \delta^{\frac{1}{3}})^3 c. \tag{6}$$

Proof (of the claim). Let $W = U^\top U$, with $U = (u_{ij})_{i,j \in [n]}$, be the Cholesky decomposition of W and denote by $u_i \in \mathbb{R}^n$, $i \in [n]$, the rows of U . It follows that

$$z^\top W z = z^\top U^\top U z = \|Uz\|^2 = \sum_{i \in [n]} (u_i^\top z)^2.$$

Let $\gamma \in (0, 1)$ and $i \in [n]$. There are two possible cases. Either $|u_i^\top y| \geq \gamma |u_i^\top z|$ and thus $|u_i^\top z| \leq \frac{1}{\gamma} |u_i^\top y|$. Or we have $|u_i^\top y| < \gamma |u_i^\top z|$. But then,

$$|u_i^\top z| = |u_i^\top y + (1 - y_\ell)u_{i\ell}| \leq |u_i^\top y| + (1 - y_\ell)|u_{i\ell}| < \gamma |u_i^\top z| + |u_{i\ell}|.$$

Hence,

$$|u_i^\top z| < \frac{1}{1-\gamma} |u_{i\ell}| = \frac{1}{1-\gamma} |u_i^\top \chi_\ell|.$$

It follows that in any of the two cases we have

$$(u_i^\top z)^2 \leq \frac{1}{(1-\gamma)^2} (u_i^\top \chi_\ell)^2 + \frac{1}{\gamma^2} (u_i^\top y)^2.$$

Using that $w_{\ell\ell} \leq \delta c$, we conclude that

$$\begin{aligned} z^\top Wz &= \sum_{i \in [n]} (u_i^\top z)^2 \leq \frac{1}{(1-\gamma)^2} \sum_{i \in [n]} (u_i^\top \chi_\ell)^2 + \frac{1}{\gamma^2} \sum_{i \in [n]} (u_i^\top y)^2 \\ &= \frac{\chi_\ell^\top W \chi_\ell}{(1-\gamma)^2} + \frac{y^\top W y}{\gamma^2} \leq \frac{\delta c}{(1-\gamma)^2} + \frac{c}{\gamma^2} \\ &= \left[\frac{\delta}{(1-\gamma)^2} + \frac{1}{\gamma^2} \right] c. \end{aligned}$$

Applying standard calculus we see that the function $\gamma \mapsto \frac{\delta}{(1-\gamma)^2} + \frac{1}{\gamma^2}$ attains its minimal value $(1 + \delta^{\frac{1}{3}})^3$ at $\gamma = (1 + \delta^{\frac{1}{3}})^{-1}$. This completes the proof of the claim. \square

Using the bound (6) we proceed to derive an upper bound on the expected value of $X^\top W X$ conditioned on $X_\ell = 1$. To that end, let $N_\ell := [n] \setminus \{\ell\}$. We have

$$\begin{aligned} \mathbb{E}[X^\top W X \mid X_\ell = 1] &= \mathbb{E} \left[\sum_{i,j \in N_\ell} w_{ij} X_i X_j + 2 \sum_{i \in N_\ell} w_{i\ell} X_i + w_{\ell\ell} \right] \\ &= \sum_{\substack{i,j \in N_\ell: \\ i \neq j}} w_{ij} \mathbb{E}[X_i] \mathbb{E}[X_j] + \sum_{i \in N_\ell} w_{ii} \mathbb{E}[X_i^2] + 2 \sum_{i \in N_\ell} w_{i\ell} \mathbb{E}[X_i] + w_{\ell\ell} \\ &= \alpha^2 \sum_{\substack{i,j \in N_\ell: \\ i \neq j}} w_{ij} y_i y_j + \alpha \sum_{i \in N_\ell} w_{ii} y_i + 2\alpha \sum_{i \in N_\ell} w_{i\ell} y_i + \alpha w_{\ell\ell} + (1-\alpha)w_{\ell\ell}, \\ &\leq \alpha \left(\sum_{\substack{i,j \in N_\ell: \\ i \neq j}} w_{ij} y_i y_j + 2 \sum_{i \in N_\ell} w_{i\ell} y_i + w_{\ell\ell} \right) + \alpha \sum_{i \in N_\ell} w_{ii} y_i + (1-\alpha)w_{\ell\ell}, \\ &= \alpha z^\top W z + \alpha \sum_{i \in N_\ell} w_{ii} y_i + (1-\alpha)w_{\ell\ell}, \end{aligned} \tag{7}$$

where the inequality follows from $\alpha \in (0, 1)$. Since y is a feasible solution of (R^k) , we have

$$\sum_{i \in N_\ell} w_{ii} y_i \leq c. \tag{8}$$

Plugging (6), (8), and $w_{\ell\ell} \leq \delta c$ into (7) yields

$$\mathbb{E}[X^\top W X \mid X_\ell = 1] \leq \alpha(1 + \delta^{\frac{1}{3}})^3 c + \alpha c + (1-\alpha)\delta c = g(\alpha, \delta)c.$$

Therefore, by Markov's inequality,

$$\begin{aligned}
\mathbb{P}[X^\top W X \leq c \mid X_\ell = 1] &= 1 - \mathbb{P}[X^\top W X > c \mid X_\ell = 1] \\
&\geq 1 - \mathbb{P}\left[X^\top W X \geq \frac{\mathbb{E}[X^\top W X \mid X_\ell = 1]}{g(\alpha, \delta)} \mid X_\ell = 1\right] \\
&\geq 1 - g(\alpha, \delta).
\end{aligned}$$

This establishes Inequality (5) and completes the proof. \square

In order to maximize the approximation guarantee of Algorithm 4, we need to find $\alpha \in (0, 1)$ that maximizes $f(\alpha, \delta)$.

Lemma 9. *For every $\delta \in (0, 1)$ the function $(0, 1) \rightarrow \mathbb{R}$, $\alpha \mapsto f(\alpha, \delta)$ attains its maximum at*

$$\alpha_\delta := \frac{1 - \delta}{(r + 1) \left[1 - \delta + (1 + \delta^{\frac{1}{3}})^3\right]},$$

and it holds that

$$\lim_{\delta \rightarrow 0} f(\alpha_\delta, \delta) = \frac{1}{2(r + 1)} \left(\frac{r}{r + 1}\right)^r \geq \frac{1}{2e(r + 1)}.$$

Proof. The fact that the function $(0, 1) \rightarrow \mathbb{R}$, $\alpha \mapsto f(\alpha, \delta)$ attains its maximum value at α_δ can be verified using standard calculus. By the continuity of f and α_δ it follows that

$$\lim_{\delta \rightarrow 0} f(\alpha_\delta, \delta) = f(\alpha_0, 0) = \frac{1}{2(r + 1)} \left(\frac{r}{r + 1}\right)^r \geq \frac{1}{2e(r + 1)},$$

which completes the proof. \square

We proceed to show that for this α , the probability that the random vector $X = \text{Ber}(\alpha y)$ produced by Algorithm 4 is infeasible for (P^k) can be bounded from above by $\frac{1}{2}$.

Lemma 10. *Let y be an optimal solution to (R^k) , $\alpha \in (0, 1)$, and $X = \text{Ber}(\alpha y)$. Then $\mathbb{P}[X \text{ infeasible for } (P^k)] \leq r(\alpha^2 + \alpha)$. In particular, if $\alpha = \alpha_\delta$, then $\mathbb{P}[X \text{ infeasible for } (P^k)] \leq \frac{1}{2}$.*

Proof. Since for every $i, j \in [n]$ with $i \neq j$, X_i and X_j are stochastically independent, it holds for every $k \in [r]$ that

$$\begin{aligned}
\mathbb{E}[X^\top W^k X] &= \sum_{i, j \in [n]: i \neq j} w_{ij}^k \mathbb{E}[X_i] \mathbb{E}[X_j] + \sum_{i \in [n]} w_{ii}^k \mathbb{E}[X_i^2] \\
&= \alpha^2 \sum_{i, j \in [n]: i \neq j} w_{ij}^k y_i y_j + \alpha \sum_{i \in [n]} w_{ii}^k y_i \\
&\leq (\alpha^2 + \alpha) c^k,
\end{aligned}$$

Algorithm 5: Randomized rounding with partial enumeration

- 1 $H_\delta \leftarrow \{i \in [n] : \exists k \in [r] \text{ with } w_{ii}^k > \delta c^k\}$
 - 2 $z_\delta \leftarrow$ optimal solution of (P^k) with $x_i = 0 \forall i \in [n] \setminus H_\delta$ and $|N_1(x)| \leq \frac{r}{\delta}$ (via enumeration)
 - 3 $y_\delta \leftarrow$ approximate solution of (P^k) with $x_i = 0 \forall i \in H_\delta$ computed by randomized rounding (Algorithm 4) with $\alpha = \alpha_\delta$
 - 4 **return** $\operatorname{argmax}_{x \in \{y_\delta, z_\delta\}} P^\top x$
-

where the inequality follows from the fact that y is a feasible solution of (R^k) . Thus, Markov's inequality implies

$$\begin{aligned}
 \mathbb{P}[X \text{ not feasible}] &= \mathbb{P}[X^\top W^k X > c^k \text{ for some } k \in [r]] \\
 &\leq \sum_{k=1}^r \mathbb{P}[X^\top W^k X > c^k] \\
 &\leq \sum_{k=1}^r \mathbb{P}\left[X^\top W^k X \geq \frac{\mathbb{E}[X^\top W^k X]}{\alpha^2 + \alpha}\right] \\
 &\leq r(\alpha^2 + \alpha).
 \end{aligned}$$

Finally, for every $\delta \in (0, 1)$

$$\begin{aligned}
 r[\alpha_\delta^2 + \alpha_\delta] &\leq r[\alpha_0^2 + \alpha_0] \\
 &= r\left[\frac{1}{4(r+1)^2} + \frac{1}{2(r+1)}\right] \leq \frac{1}{2},
 \end{aligned}$$

as required. \square

To finish the proof, we show that for any constant $\delta \in (0, 1)$, any optimal solution to (P^k) contains a constant number of δ -heavy items only.

Lemma 11. *Let x^* be an optimal solution to problem (P^k) , let $\delta \in (0, 1)$, and let $H^* := \{i \in [n] : x_i^* = 1 \text{ and } i \text{ is } \delta\text{-heavy}\}$. Then $|H^*| \leq \frac{r}{\delta}$.*

Proof. Let $N_1(x^*) := \{i \in [n] : x_i^* = 1\}$ and $H := \{i \in [n] : i \text{ is } \delta\text{-heavy}\}$. Then $H^* = N_1(x^*) \cap H$. Furthermore, we have $H = \bigcup_{k=1}^r H_k$, where $H_k := \{i \in [n] : w_{ii}^k > \delta c^k\}$. Since x^* is feasible for (P^k) , for every $k \in [r]$ we have

$$c^k \geq (x^*)^\top W^k x^* \geq \sum_{i \in N_1(x^*)} w_{ii}^k \geq |N_1(x^*) \cap H_k| \delta c^k,$$

and thus $|N_1(x^*) \cap H_k| \leq \frac{1}{\delta}$. It follows that

$$|H^*| = \left| N_1(x^*) \cap \bigcup_{k=1}^r H_k \right| \leq \sum_{k=1}^r |N_1(x^*) \cap H_k| \leq \sum_{k=1}^r \frac{1}{\delta} = \frac{r}{\delta},$$

which completes the proof. \square

We are now in position to devise a randomized constant-factor approximation algorithm for Problem (P^k) . The algorithm first enumerates all solutions using only heavy items, then computes a solution with randomized rounding involving only the light items, and returns the better of the two solutions; see Algorithm 5.

Theorem 5. *For every $\bar{\varepsilon} > 0$, there are $\varepsilon > 0$ and $\delta > 0$ such that Algorithm 5 yields an $(\alpha + \bar{\varepsilon})$ -approximation for (P^k) where*

$$\alpha = \frac{1}{1 + 2(r+1)\left(\frac{r+1}{r}\right)^r} \geq \frac{1}{1 + 2e(r+1)}.$$

Proof. Let $\varepsilon > 0$ and $\delta > 0$ be arbitrary. We claim that Algorithm 5 yields a $\rho_{\varepsilon, \delta}$ -approximation where

$$\rho_{\varepsilon, \delta} := \frac{f_{\varepsilon, \delta}}{1 + f_{\varepsilon, \delta}},$$

with $f_{\varepsilon, \delta} := f(\alpha_\delta, \delta)(1 - \varepsilon)$.

Let x^* be an optimal solution of (P^k) . We distinguish two cases.

First case: $\sum_{i \in H_\delta} p_i x_i^* \geq \rho_{\varepsilon, \delta} p^\top x^*$. Then, by Lemma 11,

$$p^\top z_\delta \geq \sum_{i \in H_\delta} p_i x_i^* \geq \rho_{\varepsilon, \delta} p^\top x^*.$$

Second case: $\sum_{i \in H_\delta} p_i x_i^* < \rho_{\varepsilon, \delta} p^\top x^*$. Thus, $\sum_{i \in [n] \setminus H_\delta} p_i x_i^* \geq (1 - \rho_{\varepsilon, \delta}) p^\top x^*$. Lemma 8 yields

$$p^\top y_\delta \geq f_{\varepsilon, \delta} \sum_{i \in [n] \setminus H_\delta} p_i x_i^* \geq f_{\varepsilon, \delta} (1 - \rho_{\varepsilon, \delta}) p^\top x^* = \rho_{\varepsilon, \delta} p^\top x^*.$$

Finally, by the continuity of f and α_δ we obtain

$$\lim_{\varepsilon, \delta \rightarrow 0} \rho_{\varepsilon, \delta} = \frac{f(\alpha_0, 0)}{1 + f(\alpha_0, 0)} = \frac{1}{1 + 2(r+1)\left(\frac{r+1}{r}\right)^r},$$

which completes the proof. \square

7 Approximation Hardness

In this section, we show that packing problems with convex quadratic constraints of type (P) are APX-hard.

Theorem 6. *It is NP-hard to approximate packing problems with convex quadratic constraints by a factor of $\frac{91}{92} + \varepsilon$, for any $\varepsilon > 0$.*

Proof. We reduce from the 6-set packing problem which is NP-hard to approximate by a factor of $\frac{22}{23} + \varepsilon$ for all $\varepsilon > 0$; see Hazan et al. [11]. An instance of a 6-set packing is given by a ground set $[m]$ and a family $\mathcal{S} \subseteq 2^{[m]}$ of subsets of $[m]$ such that $|S| = 6$ for all $S \in \mathcal{S}$. A subfamily $\mathcal{S}^* \subseteq \mathcal{S}$ is a feasible solution to the 6-set packing problem if $S \cap T = \emptyset$ for all $S, T \in \mathcal{S}^*$. For a given instance of 6-set packing, and a value $k \in \mathbb{N}$ the *gap problem* is the decision problem to decide whether:

Yes: there is a solution to the 6-set packing problem of size at least k , or
No: every solution has size strictly smaller than $\frac{22}{23}k$.

For optimal sizes in the interval $[\frac{22}{23}k, k)$ any answer is admissible. The approximation hardness of 6-set packing implies that the gap problem is an NP-hard decision problem.

Let $n := |\mathcal{S}|$ and number the sets $\mathcal{S} = \{S_1, S_2, \dots, S_n\}$. Let $A = (a_{ij})_{i,j} \in \{0, 1\}^{m \times n}$ be defined as $a_{ij} = 1$ if and only if $i \in S_j$, and let $W = A^\top A$. Consider the problem

$$\begin{aligned} & \text{maximize} && \mathbf{1}^\top x \\ & \text{subject to} && x^\top W x \leq 6k, \\ & && x \in \{0, 1\}^n, \end{aligned} \tag{SP}$$

where $\mathbf{1} = (1, \dots, 1)^\top$ is the all-ones vector. We calculate

$$x^\top W x = \|Ax\|_2^2 = \sum_{i=1}^m \left(\sum_{j=1}^n a_{ij} x_j \right)^2 = \sum_{i=1}^m \left(\sum_{j \in [n]: i \in S_j} x_j \right)^2. \tag{9}$$

Suppose, we have a **Yes**-instance for the gap problem and let \mathcal{S}^* be a subset of pairwise disjoint sets of cardinality k . Then a feasible solution for (SP) is given by x^* defined as $x_j^* = 1$ if $S_j \in \mathcal{S}^*$, and $x_j^* = 0$, otherwise. Since every set S_j , $j \in [n]$, contributes at least 6 to the left hand side of the knapsack constraint (9), this solution is also optimal for (SP) and has an objective value of k .

Next, consider a **No**-instance for the gap problem, let x^* be a corresponding optimal solution of (SP), and let k' be its objective value. Since for a **No**-instance every solution of the 6-set packing problem has size strictly less than $\frac{22}{23}k$, every set that is picked beyond the first $\lfloor \frac{22}{23}k \rfloor$ sets, intersects at least once with at least one of the first $\lfloor \frac{22}{23}k \rfloor$ sets. Thus, the first $\lfloor \frac{22}{23}k \rfloor$ sets each contribute at least 6 to (9), and each of the further $k' - \lfloor \frac{22}{23}k \rfloor$ sets each contributes at least $5 + 4 - 1 = 8$ to (9). We obtain

$$\begin{aligned} 6k & \geq (x^*)^\top W x^* \\ & \geq 6 \left\lfloor \frac{22}{23}k \right\rfloor + 8 \left(k' - \left\lfloor \frac{22}{23}k \right\rfloor \right) \\ & \geq 8k' - \frac{44}{23}k \end{aligned}$$

implying $k' \leq \frac{91}{92}k$. We conclude that for a **Yes**-instance the objective value of (SP) is at least k while for a **No**-instance it is strictly less than $\frac{91}{92}k$. Therefore, the problem is NP-hard to approximate by a factor of $\frac{91}{92} + \varepsilon$ for any $\varepsilon > 0$. \square

8 Computational results

We apply our algorithms to a problem of the type described in Example 1. Specifically, we solve the welfare maximization problem for instance 134 of the

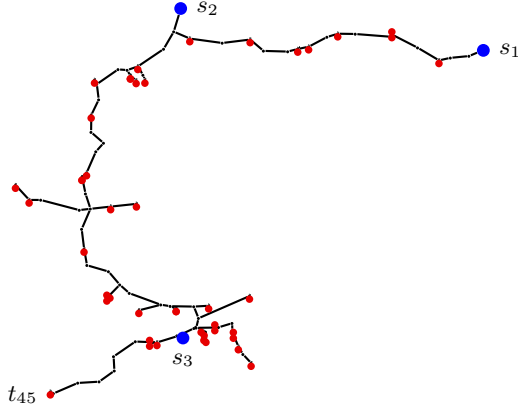


Fig. 4. The Gaslib-134 instance. Sources are shown in blue, sinks in red.

GasLib library [26]; see Figure 4 for an illustration of the network $G = (V, E)$. The instance contains upper and lower pressure bounds for every node $v \in V$ as well as all physical properties to compute the pipe resistances β_e , $e \in E$.

Sources and sinks are denoted by S and T , respectively. Every sink $t \in T$ requests a transportation of q_t units of gas to t . To ensure the robustness of the network in the sense of [15], we assume that all sinks between s_1 and s_2 are (possibly) supplied by s_1 , all sinks between s_3 and t_{45} by s_3 , and all other sinks by s_2 . Denote the set of all sinks that are (possibly) supplied by s_i by T_i , $i = 1, 2, 3$. For simplicity, we assume that the economic welfare is proportional to the amount of transported gas. That is, there is a constant $\theta > 0$ such that for every sink $t \in T$ the economic welfare p_t of transporting q_t units of gas to t equals θq_t .

Our goal is to choose a welfare-maximal subset of transportations that can be satisfied simultaneously while the pressures at the first sink s_1 and the last source t_{45} are within their feasible interval. To that end, let \bar{E} denote the path from s_1 to t_{45} , and for every $t \in T_i$ denote by E_t the set of edges on the unique path from s_i to t , $i = 1, 2, 3$. Let $p = (p_t)_{t \in T}$, $W = (w_{t,t'})_{t,t' \in T}$, with $w_{t,t'} = \sum_{e \in \bar{E} \cap E_t \cap E_{t'}} \beta_e q_t q_{t'}$, and let $c = \bar{\pi}_{s_1} - \underline{\pi}_{t_{45}}$, where for a node $v \in V$, $\bar{\pi}_v$ and $\underline{\pi}_v$ denote the upper and lower bound on the squared pressure at v , respectively. Finally, let $x = (x_t)_{t \in T} \in \{0, 1\}^T$, where $x_t = 1$ if and only if sink t is supplied. Then, the welfare-maximization problem can be formulated as (P); see Example 1.

The GasLib-134 instance contains 1234 different scenarios, where for each scenario demands \hat{q}_t are given for every sink $t \in T$. In order to make the optimization problem non-trivial we increase the node demands by setting $q_t = \gamma \hat{q}_t$, for $\gamma \in \Gamma := \{5, 10, 50, 100\}$. We apply the golden ratio algorithm, the greedy algorithm, and randomized rounding to the first 100 scenarios. For each scenario we consider every $\gamma \in \Gamma$. Each of the three algorithms is executed in three dif-

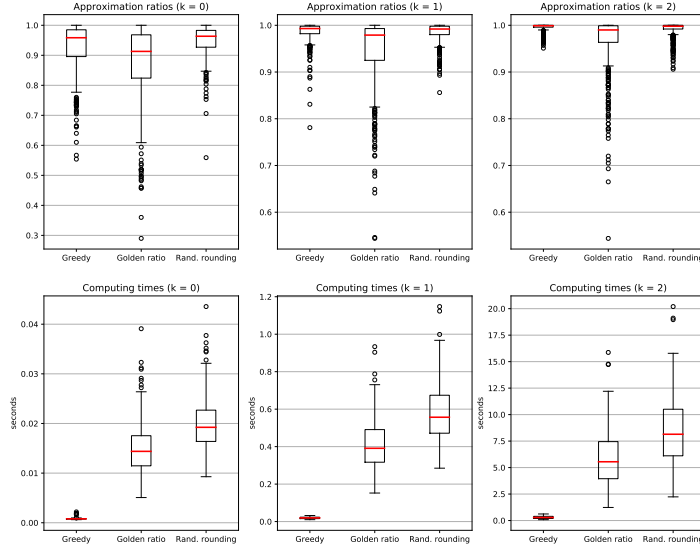


Fig. 5. Approximation ratios (top row) and computation times (bottom row) of the three algorithms when executed with partial enumeration of $k = 0, 1, 2$ elements. The red line indicates the median.

ferent versions, one without partial enumeration, one with partial enumeration with one initial item, and one with partial enumeration with two initial items.

We run randomized rounding with α chosen uniformly at random from $[0, 1]$. Instead of a single feasible realization, we generate 100 feasible realizations of $\text{Ber}(\phi y)$ and return the one with the highest profit. With the golden ratio algorithm, instead of scaling an optimal solution y of (R_1) by ϕ in order to obtain a feasible solution of (R_2) , we scale it by the largest number $\lambda \in [\phi, 1]$ such that λy is feasible for (R_2) . We find this number λ using binary search. In addition, we use the improvements described in Remark 1.

The result of each algorithm is compared to an optimal solution computed with a standard MIP solver applied to the following MIP

$$\begin{aligned}
 & \text{maximize} && p^\top x \\
 & \text{subject to} && \sum_{i=1}^n z_i \leq c, \\
 & && z_i \geq \sum_{j=1}^n w_{ij}(x_i + x_j - 1) \quad \text{for all } i \in [n], \\
 & && z_i \geq 0 \quad \text{for all } i \in [n], \\
 & && x \in \{0, 1\}^n,
 \end{aligned}$$

which can be shown to be equivalent to (P) .

	No enumeration		1-enumeration		2-enumeration	
	Mean	SD	Mean	SD	Mean	SD
Greedy	0.925	0.0837	0.985	0.0228	0.996	0.0079
Golden ratio	0.875	0.1288	0.944	0.0773	0.962	0.0639
Rand. rounding	0.948	0.0504	0.984	0.0220	0.991	0.0160

Table 1. Mean and standard deviation (SD) of the approximation ratio of the greedy algorithm, the golden ratio algorithm, and randomized rounding. Each algorithm has been executed without partial enumeration (left), with partial enumeration with one initial item (middle), and with partial enumeration with two initial items (right).

The computations are executed on a 6-core AMD Phenom II X6 1090T processor with 3.3 GHz. The code is implemented in Python 3.6 and we use the SLSQP algorithm of the SciPy `optimize` package to solve the convex relaxation (R_1). The results are shown as box plots in Figure 5 and in Table 1.

We observe that the greedy algorithm on average achieves the best approximation ratios when combined with partial enumeration. At the same time it runs approximately 20 times faster than the golden ratio algorithm and randomized rounding. The slower running time of these two algorithms is due to the fact that they rely on solving the convex relaxation first. The approximation ratio of all three algorithms is on average much higher than their proven worst case lower bounds. However, the quality of the solutions produced by the golden ratio algorithm is subject to strong fluctuations. By running the algorithm with partial enumeration with three initial items we could guarantee a ratio of at least ϕ for every instance, as was shown in Theorem 1.

References

1. Gagan Aggarwal and Jason D. Hartline. Knapsack auctions. In *Proc. 17th Annu. ACM-SIAM Sympos. Discrete Algorithms (SODA)*, pages 1083–1092, 2006.
2. Nikhil Bansal, Tracey Kimbrel, and Kirk Pruhs. Dynamic speed scaling to manage energy and temperature. In *Proc. 45th Annu. IEEE Sympos. Foundations Comput. Sci. (FOCS)*, pages 520–529, 2004.
3. Abraham Berman and Naomi Shaked-Monderer. *Completely positive matrices*. World Scientific Publishing, 2003.
4. Patrick Briest, Piotr Krysta, and Berthold Vöcking. Approximation techniques for utilitarian mechanism design. *SIAM J. Comput.*, 40:1587–1622, 2011.
5. Chi-Kin Chau, Khaled M. Elbassioni, and Majid Khonji. Truthful mechanisms for combinatorial allocation of electric power in alternating current electric systems for smart grid. *ACM trans. Econ. Comput.*, 5:Art. nr. 7, 2016.
6. Khaled M. Elbassioni and Trung T. Nguyen. Approximation algorithms for binary packing problems with quadratic constraints of low cp-rank decompositions. *Discrete Applied Mathematics*, 230:56–70, 2017.

7. Cees M. Fortuin, Pieter W. Kasteleyn, and Jean Ginibre. Correlation inequalities on some partially ordered sets. *Communications in Mathematical Physics*, 22(2):89–103, 1971.
8. Giorgio Gallo, Peter L. Hammer, and Bruno Simeone. Quadratic knapsack problems. *Math. Programming Study*, 12:132–149, 1980.
9. Allan Gibbard. Manipulation of voting schemes: a general result. *Econometrica*, 41:587–601, 1973.
10. Johan Håstad. Clique is hard to approximate within $n^{1-\epsilon}$. *Acta Mathematica*, 182(1):105–142, 1999.
11. Elad Hazan, Shmuel Safra, and Oded Schwartz. On the complexity of approximating k -dimensional matching. In S. Arora, K. Jansen, J. D. P. Rolim, and A. Sahai, editors, *Proc. 7th Int. Workshop Approximation Algorithms for Combinatorial Optim. (APPROX)*, volume 2764 of *Lecture Notes in Computer Science*, pages 83–97, 2003.
12. Oscar H. Ibarra and Chul E. Kim. Fast approximation algorithms for the knapsack and sum of subsets problems. *J. ACM*, 22:463–468, 1975.
13. Sandy Irani and Kirk R. Pruhs. Algorithmic problems in power management. *SIGACT News*, 36(2):63–76, 2005.
14. Richard M. Karp. Reducibility among combinatorial problems. In R. E. Miller, J. W. Thatcher, and J. D. Bohlinger, editors, *Complexity of Computer Computations*, The IBM Research Symposia Series. Springer, Boston, MA, 1972.
15. Max Klimm, Marc Pfetsch, Rico Raber, and Martin Skutella. On the robustness of potential-based flow networks. Technical report, 2019.
16. Mikhail K Kozlov, Sergei P Tarasov, and Leonid G Khachiyan. The polynomial solvability of convex quadratic programming. *USSR Computational Mathematics and Mathematical Physics*, 20(5):223–228, 1980.
17. Kevin A. McCabe, Stephen J. Rassenti, and Vernon L. Smith. Designing ‘smart’ computer-assisted markets: An experimental auction for gas networks. *European Journal of Political Economy*, 5:259–283, 1989.
18. Ahuva Mu’alem and Noam Nisan. Truthful approximation mechanisms for restricted combinatorial auctions. *Games Econ. Behav.*, 64:612–631, 2008.
19. Roger B. Myerson. Optimal auction design. *Math. Oper. Res.*, 6:58–73, 1981.
20. Yurii Nesterov and Arkadii Nemirovskii. *Interior-point polynomial algorithms in convex programming*, volume 13. SIAM, 1994.
21. David M. Newbery. Network capacity auctions: promise and problems. *Utilities Policy*, 11:27–32, 2002.
22. Ulrich Pferschy and Joachim Schauer. Approximation of the quadratic knapsack problem. *INFORMS J. Comput.*, 28:308–318, 2016.
23. David J. Rader Jr. and Gerhard J. Woeginger. The quadratic 0–1 knapsack problem with series-parallel support. *Oper. Res. Lett.*, 30:159–166, 2002.
24. Stephen J. Rassenti, Stanley S. Reynolds, and Vernon L. Smit. Cotenancy and competition in an experimental auction market for natural gas pipeline networks. *Econ. Theory*, 4:41–65, 1994.
25. Sartaj Sahni. Approximate algorithms for the 0/1 knapsack problem. *J. ACM*, 22(1):115–124, 1975.
26. Martin Schmidt, Denis Afmann, Robert Burlacu, Jesco Humpola, Imke Joormann, Nikolaos Kanelakis, Thorsten Koch, Djamal Oucharif, Marc E. Pfetsch, Lars Schewe, Robert Schwarz, and Mathias Sirvent. GasLib – A Library of Gas Network Instances. *Data*, 2(4):article 40, 2017.
27. Maxim Sviridenko. A note on maximizing a submodular set function subject to a knapsack constraint. *Oper. Res. Lett.*, 32:41–43, 2004.

28. Thos R. Weymouth. Problems in natural gas engineering. *Trans. Am. Soc. Mech. Eng.*, 34:185–231, 1912.
29. Adam Wierman, Lachlan L. H. Andrew, and Ao Tang. Power-aware speed scaling in processor sharing systems: Optimality and robustness. *Perform. Eval.*, 69:601–622, 2012.
30. Gerhard J. Woeginger. When does a dynamic programming formulation guarantee the existence of a fully polynomial time approximation scheme (FPTAS)? *INFORMS Journal on Computing*, 12:57–74, 2000.