Asymptotic Behaviour of the Quadratic Knapsack Problem

Joachim Schauer *

Abstract

We study subclasses of the quadratic knapsack problem, where the profits are independent random variables defined on the interval $[0, 1]$ and the knapsack capacity is proportional to the number of items (we assume that the weights are arbitrary numbers from the interval $[0, 1]$). We show asymptotically that the objective value of a very easy heuristic is not far away from the optimal solution. More specifically we show the the ratio of the optimal solution and the objective value of this heuristic almost surely tends to 1 as the size of the knapsack instance tends to infinity. As a consequence using randomly generated instances following this scheme seems to be inappropriate for studying the performance of heuristics and (to some extend) exact methods. However such instances are frequently used in the literature for this purpose. Additionally we introduce a new class of test instances for which finding a good solution is much harder. We support this by theoretical observations as well as by performing computational experiments.

Keywords: quadratic knapsack problem, asymptotic analysis, random instances

1 Introduction

In the quadratic knapsack problem (QKP) we are given a set of n items $I_1, \ldots, I_n$ each with an integer profit p_j and weight w_j. Moreover for each pair of items an additional profit p_{ij} is given which accounts for interdependencies between the involved items. We look for a subset of items whose total weight does not exceed a given capacity bound c and whose total profit is maximized. The problem can

*University of Graz, Department of Statistics and Operations Research, Universitaetsstr. 15, A-8010 Graz, Austria, *joachim.schauer@uni-graz.at*

be modelled by the well known quadratic integer programming formulation:

$$(QKP) \qquad \max \sum_{i=1}^{n} p_i x_i + \sum_{1 \le i < j \le n} p_{ij} x_i x_j \tag{1}$$

$$\text{s.t.} \sum_{i=1}^{n} w_i x_i \le c \tag{2}$$

$$x_i \in \{0,1\}, \quad i = 1, \ldots, n \tag{3}$$

Here $x_j = 1$ means that item I_j is included into the knapsack. Note that some profits p_{ij} might also be 0, meaning that the involved items do not influence each other at all.

The quadratic knapsack problem is a classical combinatorial optimization problem that was studied a lot in the literature and has many real-world applications (see e.g. Pisinger (2007), Kellerer et al. (2004, Sec.12)). The NP-hardness of QKP can be easily seen by a reduction from the maximum clique problem. When negative profits are allowed in a QKP instance Rader Jr. and Woeginger (2002) proved that the problem becomes inapproximable, however for instances containing only non-negative profits the approximability behaviour remains open due to the following connection to the *densest-k subgraph* problem: this problem asks for a k vertex induced subgraph of a given graph $G = (V, E)$ containing the maximum number of edges. This problem can be modelled as a QKP: the vertices v_i of V correspond to items I_i and all items have profits $p_i = 0$ and weights $w_i = 1$. The quadratic profit p_{ij} is set to 1 whenever $(v_i, v_j) \in E$ and otherwise to 0. The knapsack constraint is set to k. Note that the densest-k subgraph problem is infamous for its open approximability status: the existence of a $PTAS$ for this problem is still open (under the standard assumption $\mathcal{P} \neq \mathcal{NP}$) and the best known approximation algorithm of Bhaskara et al. (2010) has a performance ratio of $O(n^{\frac{1}{4}})$. Inapproximability results are only known under weaker complexity assumptions: Feige (2002) and Khot (2006) ruled out the existence of a PTAS under complexity assumptions dealing with average case hardness. Alon et al. (2011) showed that an hardness assumption on random k-AND formulas rules out the existence of any constant factor approximation algorithm. Finally Alon et al. (2011) even showed superconstant inapproximation results based on the hidden clique assumption.

On the other hand there are many papers focusing on exact algorithms and (meta-)heuristics in the literature. In fact many of them computationally show a very good performance, which in some sense contrasts the hardness of approximation results from above. Pisinger (2007) surveys the exact methods in detail. The main result of this paper indicates that this mismatch might be explained by the test instances used. The standard benchmark instances have its origin in Gallo et al. (1980) and were subsequently used in most computational papers. First a density δ is defined which corresponds to the probability that a profit p_i or p_{ij} is non-zero. Whenever a profit p_{ij} $(i < j)$ or p_i is non-zero

it is drawn uniformly at random from the interval $[1, 100]$. The weights w_i are drawn uniformly at random from the interval $[1, 50]$. Finally the capacity is drawn uniformly at random from the interval $[0, \sum w_i]$ (in more recent papers this interval was changed to $[50, \sum w_i]$).

The main result of this paper is that QKP restricted to instances where the profits are independent and (to some extent) identically distributed random variables, the weights are arbitrary numbers from the interval $[0, 1]$ and the knapsack constraint is proportional to the number of items has the following property: the ratio of the solution value of the algorithm which greedily includes the items in non-decreasing order of their weights and the optimal solution value almost surely tends to 1. Note that the result even holds whenever the profits are defined on an interval $[0, M]$ for some constant M, since a Chernoff-Hoeffding bound similar to to the one of Theorem 1 holds in this case. Hence the instances of Gallo et al. (1980) fit into this scheme whenever the drawn capacity is linear to the number of items.

Pisinger et al. (2007) introduced other types of instances which we will discuss in Section 3. Moreover we will also introduce a class of instances for which the main result of this paper is not applicable. In Section 4 we will computationally show that they are indeed hard to solve for selected state of the art heuristic approaches from the literature.

Caprara et al. (1999) provided a fast and successful branch and bound algorithm where the calculation of the upper bounds is based on Lagrangian relaxation. Billionnet and Soutif (2004) presented an algorithm based on Lagrangian decomposition. Note that the algorithm of Caprara et al. (1999) works better for dense instances, whereas the algorithm of Billionnet and Soutif (2004) has its strength on spare instances (with small δ). Both where able to solve instances of about 400 items. Helmberg et al. (2000) presented bounds based on semidefinite programming and cutting planes. Pisinger et al. (2007) introduced an aggressive reduction strategy in order to fix some variables to its optimal solution values and demonstrated that instances of size up to 1500 variables can often be reduced significantly to instances containing about 100 items only. After the reduction strategy they used an adaption of the branch and bound algorithm of Caprara et al. (1999). Recently Fomeni et al. (2014) presented an cut and branch algorithm which is especially successful on sparse instances. Létocart et al. (2012) gave a method for improving the computation time for getting upper bounds which can be incorporated in the above approaches.

Notable meta-heuristic approaches are the genetic algorithms of Julstrom (2005) and Julstrom (2012), which showed a good performance on the standard Gallo et al. (1980) test instances. The algorithm based on tabu search and GRASP from Yang et al. (2013) seems to be the currently best performing meta heuristic for QKP: even for instances containing 2000 items the gap between the best found solution and an upper bound from Caprara et al. (1999) was small ($< 1.5\%$). A heuristic by Fomeni and Letchford (2013) which modified the classical dynamic program for the 0-1 knapsack problem together with a simple local search was

able to find the optimum for almost all standard instances of size up to 380 items.

1.1 Asymptotic results for the quadratic assignment problem

The article of Burkard and Fincke (1985) generalized an earlier result on the quadratic assignment problem to so called *generic optimization problems*. Here a set E is given and the set F of feasible solutions S is a subset of the power set of E (i.e. $F \subseteq \mathcal{P}(E)$). Moreover every element $e \in E$ gets assigned a cost $c(e)$. The sum objective function is defined in the classical way for every set S as:

$$C(S) = \sum_{e \in S} c(e)$$

Burkard and Fincke (1985) showed that whenever the costs $c(e)$ are i.i.d. random variables defined on the interval $[0, 1]$ and some additional assumptions are fulfilled then for generic optimization problems with sum objective function the ratio of the worst and best feasible solution asymptotically tends to 1 in probability. The most important additional assumptions are the following:

1. The size of every feasible solution S is the same.
2. $\exists \lambda > 0 : \lim_{n \to \infty} (\lambda |S| - \log |F|) = \infty$

Assumption 2 is crucial and this property (though not stated explicitly in this form) is also central for our problem and is needed for inequality (10). Later this result was extended to almost sure convergence by Szpankowski (1995). Similar results also hold for bottleneck objective functions.

Note however that there exists a result by Dyer et al. (1986) which states that certain branch and bound algorithms will take more than exponential time for quadratic assignment instances with random costs. Hence the results mentioned in this section as well as our result do not imply that finding the optimal solution of such a problem is easy. They just imply that finding good solutions is easy.

2 Asymptotic analysis

For deriving our main result we need the following Chernoff-Hoeffding bounds that are based on Angluin and Valiant (1979) and can be found in McDiarmid (1998, Theorem 2.3 on page 200):

Theorem 1. *Let the random variables $X_1, X_2, \ldots, X_n$ be independent with $0 \leq X_k \leq 1$ for each k. Let $S_n = \sum X_k$ and let $\mu = E(S_n)$. Then for any $0 \leq \varepsilon \leq 1$:*

$$P\left[S_n \geq (1+\varepsilon)\mu\right] \leq e^{-\frac{1}{3}\varepsilon^2\mu}$$
$$P\left[S_n \leq (1-\varepsilon)\mu\right] \leq e^{-\frac{1}{2}\varepsilon^2\mu}$$

In this section we assume that the profits of a quadratic knapsack instance are independent random variables following distributions which are defined on the interval $[0,1]$ and the weights are arbitrary numbers from $[0,1]$. This means that the weights might be random or part of the input. Moreover we assume that the knapsack capacity c is proportional to n (i.e. $c = \lambda n$ for some $\lambda \leq 1$). Note that the random variables corresponding to the profits P_i and P_{ij} are all assumed to be independent. More specifically the P_i are assumed to be i.i.d. and the P_{ij} are assumed to be i.i.d. - however the distributions might be different for the two profit classes. We moreover assume that the random variables X involved have a positive and constant expectation (i.e. $E(X) = \mu_X > 0$). With this we define the *asymptotic-$QKP(n)$* problem:

$$a\text{-}QKP(n) \qquad \max \sum_{i=1}^{n} P_i x_i + \sum_{1 \leq i < j \leq n} P_{ij} x_i x_j \tag{4}$$
$$\text{s.t.} \sum_{i=1}^{n} W_i x_i \leq \lambda n \tag{5}$$
$$x_i \in \{0,1\}, \quad i = 1, \ldots, n \tag{6}$$

In case that the weights are random variables, we define for a-$QKP(n)$ a random variable L as the maximum number of items which can be feasibly included into the knapsack. Assume that a realization of the random variables W_i of a-$QKP(n)$ is given: then the realization of L can be easily determined by ordering the items in non-decreasing order of their realized weights. The realization of L now corresponds to the index l such that $\sum_{i=1}^{l} w_i \leq \lambda n$ and $\sum_{i=1}^{l+1} w_i > \lambda n$. Here the lower case w_i denotes the realized value of the random variable W_i. $Z^A(n)$ denotes the random variable corresponding to the objective value that results by including the L lightest items and $Z^*(n)$ denotes the random variable which corresponds to the optimal solution value of the given instance.

In the following we will show asymptotically that $Z^A(n)$ and $Z^*(n)$ are not far from each other. More specifically we will show that for any positive constant δ we get:

$$\lim_{n\to\infty} P\left[\frac{Z^*(n)}{Z^A(n)} \leq (1+\delta)\right] = 1$$

We will relax a given a-$QKP(n)$ instance I by replacing the knapsack constraint with a cardinality constraint. By F_n^l we denote the set of all subsets of items of cardinality exactly l. Note that $|F_n^l| = \binom{n}{l} < 2^n$. We define the following random variables for all $S \in F_n^l$:

$$Z_l^S(n) = \sum_{i \in S} P_i + \sum_{1 \leq i < j \leq n | i,j \in S} P_{ij}$$

$$Z_l^{max}(n) = \max_{S \in F_n^l} Z_l^S(n) \qquad Z_l^{min}(n) = \min_{S \in F_n^l} Z_l^S(n)$$

It is crucial that in an a-$QKP(n)$ instance with n items at least λn items fit, hence $L \geq \lambda n$. Since $Z^*(n)$ corresponds to a solution containing $\leq L$ items with $\lambda n \leq L \leq n$ there always exits a certain index $l' \geq \lambda n$ such that the following inequality holds:

$$Z_{l'}^{max}(n) \geq Z^*(n) \geq Z^A(n) \geq Z_{l'}^{min}(n) \tag{7}$$

We denote by μ_m the minimum expected value of the random variables P_i, P_{ij} of our instance. With this and the linearity of expectation we get that for a

$$E[Z_l^S(n)] = E\left[\sum_{i \in S} P_i + \sum_{1 \leq i < j \leq n | i,j \in S} P_{ij}\right] \geq l\mu_m + \frac{l(l-1)}{2}\mu_m \geq \frac{\lambda^2 n^2}{2}\mu_m \tag{8}$$

Theorem 2. *Consider a sequence of a-$QKP(n)$ problems with random profit variables P_k and P_{ij} and arbitrary weights from $[0,1]$ and a knapsack capacity of λn ($k \in \{1, 2 \ldots, n\}$, $1 \leq i < j \leq n$ and $\lambda > 0$ and constant). Moreover assume that all random variables are defined on the interval $[0,1]$, are independent and have a positive and constant expected value. Each of the P_k and each of the P_{ij} have to be moreover identically distributed. Let $\delta > 0$ be an arbitrary positive constant. Let $Z^*(n)$ denote the random variable corresponding to the optimal solution and $Z^A(n)$ be random variable obtained by including the L lightest items. Then we get:*

$$\lim_{n \to \infty} P\left[\frac{Z^*(n)}{Z^A(n)} \leq (1+\delta)\right] = 1$$

Proof.

We will show that for all $l \geq \lambda n$ the following holds:

$$\lim_{n \to \infty} P\left[\frac{Z_l^{max}(n)}{Z_l^{min}(n)} \leq (1+\delta)\right] = 1 \tag{9}$$

One of these indices corresponds to l' from equation (7), hence equation (9) implies the theorem. For showing (9) we will choose an arbitrary positive ε and show that:

$$\lim_{n\to\infty} P\Big[Z_l^{max}(n) \geq (1+\varepsilon)E(Q_l^n)\Big] = 0 \tag{10}$$

$$\lim_{n\to\infty} P\Big[Z_l^{min}(n) \leq (1-\varepsilon)E(Q_l^n)\Big] = 0 \tag{11}$$

Here $E(Q_l^n)$ denotes the expected objective value over all knapsacks containing l items, while ignoring the capacity constraint:

$$E(Q_l^n) = \sum_{S\in F_n^l} \frac{Z_l^S(n)}{\binom{n}{l}}$$

Equations (**??**) and (**??**) mean that

$$\frac{Z_l^{max}(n)}{E(Q_l^n)} \text{ and } \frac{Z_l^{min}(n)}{E(Q_l^n)}$$

both converge in probability to 1. Hence by applying the Continuous Mapping Theorem (e.g. (Van der Vaart, 2000, Theorem 2.3)) we get that also

$$\frac{Z_l^{max}(n)}{Z_l^{min}(n)}$$

converges in probability to 1, which is equivalent to (9).

We will prove equation (**??**): let S' now be any set of l knapsack items.

$$P\Big[Z_l^{max}(n) \geq (1+\varepsilon)E(Q_l^n)\Big] = P\left[\bigvee_{S\in F_n^l} \big(Z_l^S(n) \geq (1+\varepsilon)E(Q_l^n)\big)\right] \leq$$

$$\leq \sum_{S\in F_n^l} P\Big[Z_l^S(n) \geq (1+\varepsilon)E(Q_l^n)\Big] = |F_n^l| \cdot P\Big[Z_l^{S'}(n) \geq (1+\varepsilon)E(Q_l^n)\Big] \leq$$

$$\leq |F_n^l| \cdot e^{-\frac{1}{3}\varepsilon^2 E(Q_l^n)} \leq |F_n^l| \cdot e^{-\frac{1}{3}\varepsilon^2 \frac{\lambda^2 n^2}{2}\mu_m}$$

The first inequality uses the union bound, the second inequality uses the Chernoff-Hoeffding bound from Theorem 1 and the third inequality comes from equation (8). Note that the random variables P_k and P_{ij} are independent from the set of selected items S.

But now we have that:

$$\lim_{n\to\infty} \left(|F_n^l| \cdot e^{-\frac{1}{3}\varepsilon^2 \frac{\lambda^2 n^2}{2}\mu_m}\right) \leq \lim_{n\to\infty} \left(2^n \cdot \left(e^{-\frac{\varepsilon^2\lambda^2\mu_m}{6}}\right)^{n^2}\right) = 0 \tag{12}$$

Equation **??** follows in an analogous way and both together imply (9) and the Theorem.

□

By applying the Borel-Cantelli Lemma we get that the above result also holds almost surely.

Theorem 3. *Consider a sequence of a-QKP(n) problems with random profit variables P_k and P_{ij} and arbitrary weights from $[0,1]$ and a knapsack capacity of λn ($k \in \{1,2\ldots,n\}$, $1 \leq i < j \leq n$ and $\lambda > 0$ and constant). Moreover assume that all random variables are defined on the interval $[0,1]$, are independent and have a positive and constant expected value. Each of the P_k and each of the P_{ij} have to be moreover identically distributed. Let $\delta > 0$ be an arbitrary positive constant. Let $Z^*(n)$ denote the random variable corresponding to the optimal solution and $Z^A(n)$ be random variable obtained by including the L lightest items. Then we get:*

$$P\left[\lim_{n\to\infty} \frac{Z^*(n)}{Z^A(n)} = 1\right] = 1$$

Proof.

Again by the Continuous Mapping Theorem it is sufficient to show that:

$$P\Big[\lim_{n\to\infty} \frac{Z_l^{max}(n)}{E(Q_l^n)} = 1\Big] = 1 \tag{13}$$

$$P\Big[\lim_{n\to\infty} \frac{Z_l^{min}(n)}{E(Q_l^n)} = 1\Big] = 1 \tag{14}$$

By the Borel-Cantelli Lemma (11) (resp. (12)) hold whenever (13) (resp. (14)) hold:

$$\sum_{n=1}^{\infty} P[Z_l^{max}(n) \geq (1+\varepsilon)E(Q_l^n)\Big] < \infty \tag{15}$$

$$\sum_{n=1}^{\infty} P[Z_l^{max}(n) \leq (1-\varepsilon)E(Q_l^n)\Big] < \infty \tag{16}$$

We focus on (13). By the proof of Theorem 2 we know that:

$$\sum_{n=1}^{\infty} P[Z_l^{max}(n) \geq (1+\varepsilon)E(Q_l^n)\Big] \leq \sum_{n=1}^{\infty} \left(2^n \cdot \left(e^{-\frac{\varepsilon^2\lambda^2\mu_m}{6}}\right)^{n^2}\right) < \infty$$

The convergence of the last sum can for example be checked by applying the ratio test. The proof of (14) works analogously.

□

Remark 1. Both Theorems of this section are still valid in case that the profits are drawn from distributions defined on an interval $[0, M]$ for a fixed M. This follows by the fact that in this case Chernoff-Hoeffding bounds similar to Theorem 1 exist.

3 QKP-Instances

In this section we review classes of test instances used in the literature for QKP. We moreover argue whether they are captured by Theorem 3 and furthermore introduce a new class of test instances, which should be computationally hard for heuristical as well as exact QKP algorithms.

3.1 Classical instances (*CI*) of Gallo et al. (1980)

As already noted in the introduction, instances of Gallo et al. (1980) are frequently used for computational studies in the literature. Here the quadratic profits P_{ij} are generated in the following way: first a Bernoulli distribution with parameter δ is used for deciding whether P_{ij} is present in the instance. If P_{ij} is present then its actual value is determined by drawing an integer from $[1, 100]$ uniformly at random. Hence the value of P_{ij} corresponds to the product of the outcome of the two random variables involved. Since all the Bernoulli random variables are i.i.d. and all the variables from the uniform distribution are i.i.d. too, also their product is i.i.d.. Since in the classical instances all linear profits P_i are present it is sufficient to directly use the fact that they are modelled as i.i.d. random variables. Moreover the weights w_i are drawn uniformly at random from the interval $[1, 50]$. Finally the capacity is drawn uniformly at random from the interval $[50, \sum w_i]$. Hence by Remark 1 the Theorems 2 and 3 are applicable as long as the knapsack capacity is proportional to n.

3.2 Densest-k subgraph (*DKS*) instances of Pisinger et al. (2007)

Here an Erdős-Rényi random graph $G(n, p)$ is given and the goal is to find the densest-k subgraph of this graph (p has been chosen to be $\in \{0.25, 0.5, 0.75, 0.9\}$) and k is randomly chosen from $\{2, 3, \ldots, n-2\}$. Again for large enough k theses instances are of the type we study in this paper. In our computational experiments of Section 4 we did not choose k uniformly at random instead we used a knapsack capacity of $0.1n$ and $0.3n$, where n is the number of items of the instance. This was done in order to guarantee that c is proportional to n.

3.3 Dispersion problem instances introduced by Pisinger et al. (2007)

Here one wants to locate q facilities at n possible locations in order to maximize the sum of all the distances between the facilities at their chosen locations. Pisinger et al. (2007) gave the following quadratic formulation:

$$\max \sum_{i \in N} \sum_{j \in N} d_{ij} x_i x_j \tag{17}$$

$$\text{s.t.} \sum_{j \in N} x_j = q \tag{18}$$

$$x_j \in \{0, 1\} \tag{19}$$

Clearly this problem is a QKP. Pisinger et al. (2007) introduced several variants of how the distances d_{ij} are determined (d_{ii} is always 0):

- the n locations are randomly located in a rectangle of size 100×100 and the distance is the Euclidean one
- additionally each location i gets a weight $\alpha_i \in [5, \ldots, 10]$ and the distance between i and j is then determined by the Euclidean one times $\alpha_i \alpha_j$
- each distance d_{ij} with $i < j$ is drawn from an exponential distribution with mean 50 and $d_{ij} = d_{ji}$

Finally q is uniformly drawn from $\{2, 3, \ldots, n-2\}$. Moreover Pisinger et al. (2007) modified these instances by introducing weights from $[1, \ldots, 100]$ for each location exchanging the cardinality constraint by a knapsack constraint with $c = \frac{1}{2} \sum_{i=1}^{N} w_i$. Note that these instances are all not captured by our framework.

It turned out that the densest-k subgraph instances were computationally the hardest ones for the approach of Pisinger et al. (2007). They explain this by the fact that the costs and weights in densest-k subgraph instances are very homogeneous which causes problems for their aggressive reduction strategy.

3.4 New Instances

Hidden Clique Instances

The superconstant hardness result of Alon et al. (2011) relies on the hidden clique problem: in a given realization of $G(n, \frac{1}{2})$ a set of l vertices is chosen uniformly at random and turned into a complete graph. It is well known that the size of the maximum clique of a $G(n, \frac{1}{2})$ is about $2 \log_2(n)$. It is now assumed

that finding the hidden clique in polynomial time when $l = n^a$ with $a < \frac{1}{2}$ is impossible (see Alon et al. (2011) for a detailed discussion). Note that when $a \geq \frac{1}{2}$ the problem is polynomial time solvable by Alon et al. (1998). A hidden clique instance in the QKP setting now consists of choosing a realization of $G(n, \frac{1}{2})$, planting a clique of size $\lfloor\sqrt{n}\rfloor$[‡] in it and setting the knapsack capacity to $\lfloor\sqrt{n}\rfloor$. The advantages of these instances are the following: there is theoretical evidence that solving such instances is hard; they are not capture by Theorem 2 and 3; the optimal solution value is known (with overwhelming probability) to be equal to

$$\frac{1}{2}(\lfloor\sqrt{n}\rfloor)(\lfloor\sqrt{n}\rfloor - 1)$$

4 Computational Experiments

In this section we give computational results for instances described in Section 3. We tested the algorithms of Julstrom (2005) (JUL), Fomeni and Letchford (2013) (called FL), as well as the greedy algorithm (TG) used for the theoretical results of Section 2 and an adaption (OG) of Julstrom (2005) specially tuned for getting good results in a very short amount time[§]. The last algorithm was implemented for solving the real world problem of Pferschy et al. (2015). The computational experiments were all performed on an Intel Core i5-3470 CPU with 3.20GHz and 16 GB RAM running Ubuntu Linux. The codes of the algorithms were compiled using GCC (version 4.6.3.).

We tested the algorithms with instances *CI* from Section 3.1, *DKS* from Section 3.2 and *HI* from Section 3.4. able 1 describes the different versions of instances *CI* and *DKS* used.

instance type	parameters	capacity
CI (Section 3.1)	$\delta = 25, 50, 75, 100$	see Section 3.1
DKS (Section 3.2)	$p = 0.25, 0.5, 0.75, 0.9$	$c = 0.1n, 0.3n$

Table 1: Test Instances wth Parameters

For each instance type that we used for testing we randomly generated 10 instances. The genetic algorithm of Julstrom (2005) computes $10n$ generations, in case of $n = 800$ items this resulted in an average runtime of 1411 seconds for the instances of type CI, hence we set the maximum runtime of our adapted genetic algorithm to 1400 seconds. In fact for our adapted implementation we calculated the best found solution after 10, 15, 30, 60, 120 and 1400 seconds. The algorithm of Fomeni and Letchford (2013) runs in $O(n^2c)$ time. For a fair

[‡]This setting gives large hidden cliques compared to the expected cliques of size $2\log_2(n)$, although from a theoretical perspective $a \not< \frac{1}{2}$.

[§]Unfortunately we did not get the code (resp. binary) of Yang et al. (2013).

comparison we only report the solutions for OG that were found after running the algorithm for approximately as long as JUL.

Table 2 reports the results for large instances of type *CI*. The column 'best' marks the algorithm that found the best solution and the column 'gap' indicates how much the other algorithms deviate from the best found solution (3 decimals used). The column 'av. solution value' reports the average value over the 10 randomly generated instances and the average solution time is reported with respect to one decimal. The gap of the solutions of TG compared to the best found solution is already small for instances of size $n = 800$[‡]. Moreover it turns out that the gap becomes smaller for denser instances - which perfectly matches the theoretical considerations.

algorithm	param	av. time (sec)	av. solution value	best	gap
FL	$\delta = 0.25, n = 800$	264.3	2606699.7		0.002%
JUL	$\delta = 0.25, n = 800$	1387.5	2606763.1	∘	
OG	$\delta = 0.25, n = 800$	1403.2	2606634.6		0.005%
TG	$\delta = 0.25, n = 800$	0	2586000.6		0.803%
FL	$\delta = 0.5, n = 800$	250	4957186.8		0.001%
JUL	$\delta = 0.5, n = 800$	1394.3	4956687.2		0.011%
OG	$\delta = 0.5, n = 800$	1403.4	4957241.2	∘	
TG	$\delta = 0.5, n = 800$	0	4937327.2		0.403%
FL	$\delta = 0.75, n = 800$	211.9431	6603402.3	∘	
JUL	$\delta = 0.75, n = 800$	1421.8	6603402.0		0.000%
OG	$\delta = 0.75, n = 800$	1403.5	6602235.9		0.018%
TG	$\delta = 0.75, n = 800$	0	6584901.3		0.281%
FL	$\delta = 1, n = 800$	195.7	8023573.5	∘	
JUL	$\delta = 1, n = 800$	1441.9	8023439.3		0.002%
OG	$\delta = 1, n = 800$	1403.4	8021291.7		0.028%
TG	$\delta = 1, n = 800$	0	8009417.1		0.177%

Table 2: Computational Results for CI with $n = 800$

Table 3 reports the solutions for large DKS instances. Here the algorithm FL already performs worse than the two genetic algorithms. TG in this case selects a random subset of 80 and 240 items respectively (all items have weight 1). One can observe that the gap becomes better for TG with increasing densities and increasing knapsack capacities, which again is in line with the theoretical results of Section 2. Note that Pisinger et al. (2007) already identified these instances as very hard instances for their exact approach[§].

Finally Table 4 reports the solution for the hidden clique instances. TH reports the size of the hidden clique and now the column 'gap' takes this value as

[‡]The algorithm of Caprara et al. (1999) did not find exact solutions in a time limit of 10 hours.

[§]For DKS instances of size $n = 200$ the algorithm of Caprara et al. (1999) did not find exact solutions in a time limit of 10 hours.

reference. Moreover we report the best found solution value of OG for several running times. Their development gives a strong hint that this genetic algorithm is not able to improve its solution quality much by using longer running times. It seems that it gets stuck at the local optimum which is near its starting solution. All algorithms have troubles in finding the hidden clique, especially at $n = 800$. TG finds a solution with gap around 100%. This is reasonable, since it randomly chooses a set of $\lfloor\sqrt{n}\rfloor$ items, where each edge occurs with probability $\approx \frac{1}{2}$. In contrast to the instances of type CI and DKS our own adoption seems to be the best performing meta-heuristic. The reason might be the following difference in generating the start population. JUL computes densities d_i for all items i by:

$$d_i = \frac{p_i + \sum_{j|(i,j)\in E} p_{ij}}{w_i}$$

Then the feasible solutions of the starting generation are generated by a tournament-like process. Two items are chosen randomly, then a random coin flip which is biased towards the item with higher density decides which of the two items is included into the knapsack (if it fits).

OG uses a different method for computing densities $\bar{d}_i$ which are then used in a similar tournament :

$$\bar{d}_i = \frac{p_i}{w_i} + \sum_{j|(i,j)\in E} \frac{p_j + p_{ij}}{w_i + w_j}$$

Moreover note that OG was especially tuned for finding good solutions in less than 3 seconds.

Acknowledgements

We thank the authors of Caprara et al. (1999), Julstrom (2005) and Fomeni and Letchford (2013) for providing the source codes for the computational experiments of Section 4. We moreover thank Ulrich Pferschy for valuable comments and Rostislav Staněk and Enrico Grande for helping to perform the computational experiments. The research was funded by the Austrian Science Fund (FWF): P23829.

References

N. Alon, M. Krivelevich, and B. Sudakov. Finding a large hidden clique in a random graph. *Random Structures and Algorithms*, 13(3-4):457–466, 1998.

N. Alon, S. Arora, R. Manokaran, D. Moshkovitz, and O. Weinstein. Inapproximabilty of Densest k-Subgraph from Average Case Hardness . Technical report, 2011.

D. Angluin and L.G. Valiant. Fast probabilistic algorithms for hamiltonian circuits and matchings. *Journal of Computer and System Sciences*, 18(2):155 – 193, 1979.

A. Bhaskara, M. Charikar, E. Chlamtac, U. Feige, and A. Vijayaraghavan. Detecting high log-densities: an o (n 1/4) approximation for densest k-subgraph. In *Proceedings of the forty-second ACM symposium on Theory of computing*, pages 201–210. ACM, 2010.

A. Billionnet and É. Soutif. An exact method based on lagrangian decomposition for the 0–1 quadratic knapsack problem. *European Journal of operational research*, 157(3):565–575, 2004.

R.E. Burkard and U. Fincke. Probabilistic asymptotic properties of some combinatorial optimization problems. *Discrete Applied Mathematics*, 12(1):21 – 29, 1985.

A. Caprara, D. Pisinger, and P. Toth. Exact solution of the quadratic knapsack problem. *INFORMS Journal on Computing*, 11(2):125–137, 1999.

M.E. Dyer, A.M. Frieze, and C. McDiarmid. On linear programs with random costs. *Mathematical Programming*, 35(1):3–16, 1986.

U. Feige. Relations between average case complexity and approximation complexity. In *STOC '02: Proceedings of the 34th Symposium on Theory of Computing*, pages 534–543. ACM, 2002.

F.D. Fomeni and A.N. Letchford. A dynamic programming heuristic for the quadratic knapsack problem. *INFORMS Journal on Computing*, 26(1):173–182, 2013.

F.D. Fomeni, K. Kaparis, and A.N. Letchford. A cut-and-branch algorithm for the quadratic knapsack problem, 2014. URL `http://www.optimization-online.org/DB_HTML/2014/06/4379.html`.

G. Gallo, P.L. Hammer, and B. Simeone. Quadratic knapsack problems. In M.W. Padberg, editor, *Combinatorial Optimization*, volume 12 of *Mathematical Programming Studies*, pages 132–149. Springer Berlin Heidelberg, 1980.

C. Helmberg, F. Rendl, and R. Weismantel. A semidefinite programming approach to the quadratic knapsack problem. *Journal of Combinatorial Optimization*, 4(2):197–215, 2000.

B.A. Julstrom. Greedy, genetic, and greedy genetic algorithms for the quadratic knapsack problem. In *Proceedings of the 7th annual conference on Genetic and evolutionary computation*, pages 607–614. ACM, 2005.

B.A. Julstrom. Naive and heuristic permutation-coded genetic algorithms for the quadratic knapsack problem. In *Proceedings of the 14th annual conference companion on Genetic and evolutionary computation*, pages 1465–1466. ACM, 2012.

H. Kellerer, U. Pferschy, and D. Pisinger. *Knapsack Problems*. Springer, 2004.

S. Khot. Ruling Out PTAS for Graph Min-Bisection, Dense k-Subgraph, and Bipartite Clique. *SIAM Journal on Computing*, 36(4):1025–1071, 2006.

L. Létocart, A. Nagih, and G. Plateau. Reoptimization in lagrangian methods for the 0–1 quadratic knapsack problem. *Computers & Operations Research*, 39(1):12 – 18, 2012. Special Issue on Knapsack Problems and Applications.

C. McDiarmid. Concentration. In Michel Habib, Colin McDiarmid, Jorge Ramirez-Alfonsin, and Bruce Reed, editors, *Probabilistic Methods for Algorithmic Discrete Mathematics*, volume 16 of *Algorithms and Combinatorics*, pages 195–248. Springer Berlin Heidelberg, 1998.

U. Pferschy, J. Schauer, and G. Maier. A quadratic knapsack model for optimizing the media mix of a promotional campaign. In Eric Pinson, Fernando Valente, and Begoña Vitoriano, editors, *Operations Research and Enterprise Systems*, volume 509 of *Communications in Computer and Information Science*, pages 251–264. Springer International Publishing, 2015.

D. Pisinger. The quadratic knapsack problem - a survey. *Discrete Applied Mathematics*, 155:623–648, 2007.

D. Pisinger, A.B. Rasmussen, and R. Sandvik. Solution of large quadratic knapsack problems through aggressive reduction. *INFORMS Journal on Computing*, 19:280–290, 2007.

D.J. Rader Jr. and G.J. Woeginger. The quadratic 0-1 knapsack problem with series-parallel support. *Operations Research Letters*, 30:159–166, 2002.

W. Szpankowski. Combinational optimization problems for which almost every algorithm is asymptotically optimal. *Optimization*, 33(4):359–367, 1995.

A. W Van der Vaart. *Asymptotic statistics*, volume 3. Cambridge university press, 2000.

Z. Yang, G. Wang, and F. Chu. An effective grasp and tabu search for the 0–1 quadratic knapsack problem. *Computers & Operations Research*, 40(5): 1176–1185, 2013.

algorithm	param	av. time (sec)	av. solution value	best	gap
FL	$\delta = 0.25, c = 80$	1.2	1191.2		8.68%
JUL	$\delta = 0.25, c = 80$	1709.4	1291.0		0.28%
OG	$\delta = 0.25, c = 80$	1400.9	1294.6	∘	
TG	$\delta = 0.25, c = 80$	0	789.4		64.00%
FL	$\delta = 0.5, c = 80$	1.2	2025.6		5.02%
JUL	$\delta = 0.5, c = 80$	1709.0	2127		0.01%
OG	$\delta = 0.5, c = 80$	1401.1	2127.2	∘	
TG	$\delta = 0.5, c = 80$	0	1578.5		34.76%
FL	$\delta = 0.75, c = 80$	1.2	2735.3		2.87%
JUL	$\delta = 0.75, c = 80$	1709.6	2813.7	∘	
OG	$\delta = 0.75, c = 80$	2400.2	2809.9		0.14%
TG	$\delta = 0.75, c = 80$	0	2364		19.02%
FL	$\delta = 0.9, c = 80$	1.2	3073.7		1.39%
JUL	$\delta = 0.9, c = 80$	1709.5	3116.3	∘	
OG	$\delta = 0.9, c = 80$	1401.0	3114.7		0, 05%
TG	$\delta = 0.9, c = 80$	0.0	2844.2		9.57%
FL	$\delta = 0.25, c = 240$	3.7	8796.9		2.07%
JUL	$\delta = 0.25, c = 240$	1619.6	8971.7		0.08%
OG	$\delta = 0.25, c = 240$	1401.4	8979.3	∘	
TG	$\delta = 0.25, c = 240$	0	7191.4		24.86%
FL	$\delta = 0.5, c = 240$	3.8	15888.3		3.05%
JUL	$\delta = 0.5, c = 240$	1619.4	16373	∘	
OG	$\delta = 0.5, c = 240$	1401.8	16372.8		0.001%
TG	$\delta = 0.5, c = 240$	0	14331.8		14.24%
FL	$\delta = 0.75, c = 240$	3.7	22843.5		1.84%
JUL	$\delta = 0.75, c = 240$	1619.4	23263.3	∘	
OG	$\delta = 0.75, c = 240$	1401.6	23256.6		0.029%
TG	$\delta = 0.75, c = 240$	0	21533		8.04%
FL	$\delta = 0.9, c = 240$	3.7	26715.5		0.96%
JUL	$\delta = 0.9, c = 240$	1620.3	26972.1	∘	0.008%
OG	$\delta = 0.9, c = 240$	1401.5	26970		
TG	$\delta = 0.9, c = 240$	0.0	25816.2		4.48%

Table 3: Computational Results for DKS with $n = 800$

algorithm	param	av. time (sec)	av. solution value	gap
TH	$n = 800$		378	
FL	$n = 800$	0.61	298.2	26.76%
JUL	$n = 800$	1735.3	325.1	16.27%
OG	$n = 800$	1400.7	356.1	6.15%
OG	$n = 800$	60.6	356.1	6.15%
OG	$n = 800$	10.8	355.4	6.36%
TG	$n = 800$	0	193.5	95.35%
TH	$n = 600$		276	
FL	$n = 600$	0.2	222.5	24.04%
JUL	$n = 600$	508.0	263.5	4.74%
OG	$n = 600$	1400.4	271.7	1.58%
OG	$n = 600$	10.4	271.7	1.58%
TG	$n = 600$	0	136.1	102.79%
TH	$n = 400$		190	
FL	$n = 400$	0.1	157.3	20.79%
JUL	$n = 400$	113.0	175.4	8.32%
OG	$n = 400$	1400.1	184.2	3.15%
OG	$n = 400$	120.1	183.6	3.49%
OG	$n = 400$	60.13	181.8	4.51%
OG	$n = 400$	10.1	181.8	4.51%
TG	$n = 400$	0	95.9	98.12%
TH	$n = 200$		91	
FL	$n = 200$	0	78.9	15.34%
JUL	$n = 200$	7.4	85.4	6.56%
OG	$n = 200$	315.8[§]	88.3	3.06%
OG	$n = 200$	10	88	3.41%
TG	$n = 200$	0	47	93.62%

Table 4: Computational Results for HI

[§]The algorithm reached its maximum of 50000 generations.