QKP TIG Challenge Description

TIG Labs | Aoibheann Murray

December 1, 2025

1 Problem Description and Formulation

The *Quadratic Knapsack Problem (QKP)* is a classical NP-hard combinatorial optimisation problem in which a subset of items must be selected to maximise profit while respecting a knapsack capacity constraint. Each item i has a weight w_i and a *linear* profit p_i, while each pair of items (i, j) contributes an *interaction* profit $p_{ij} = p_{ji}$. The objective is to select a feasible subset of items whose total value is maximised.

Formally, the QKP is defined over:

- a set of n items indexed by $i = 1, \dots, n$,
- item weights w_i,
- linear profits p_i,
- symmetric quadratic interaction profits p_{ij},
- a knapsack with capacity C.

The optimisation problem is:

$$\max \left(\sum_{i=1}^{n} p_i x_i \; + \sum_{1 \le i < j \le n} p_{ij} x_i x_j \right)$$

subject to:

$$\sum_{i=1}^{n} w_i x_i \le C, \qquad x_i \in \{0, 1\}.$$

A feasible solution must satisfy:

- the total weight of selected items does not exceed capacity;
- quadratic interactions contribute value only when both items in a pair are selected;
- each item is either included or excluded (binary).

2 Random Instance Generation

The instance-generation procedure follows established academic benchmarks originating from Gallo et al. [1] and widely adopted in subsequent QKP research [2, 3, 4, 5, 6, 7]. Our goal is to generate realistic, reproducible, and scientifically grounded instances that align with academic benchmarks.

2.1 Instance Attributes

Item Weights

Item weights are drawn independently:

$$w_i \sim \mathrm{UD}[1, 50], \qquad i = 1, \dots, n.$$

Linear and Quadratic Profits

Linear and quadratic profits, p_i and p_{ij} $(= p_{ji})$, are independently nonzero with a density d. When nonzero, they are drawn uniformly at random from the interval $[1, 100]$, and are set to zero otherwise.

Density

Four density values are commonly considered in the literature: $d = 0.25, 0.50, 0.75, 1.00$. Currently, density is fixed at $d = 0.25$ as lower-density instances have been noted to be more challenging and have become an area of interest in some recent academic studies [8][9].

Knapsack Capacity

Capacity is currently fixed at:

$$C = \frac{1}{2} \sum_{i=1}^{n} w_i.$$

In standard QKP instances [1], capacity C is chosen randomly from the interval $[50, \sum_{i=1}^{n} w_i]$.

3 Instance Collections

This section describes the generation procedures used for all synthetic instance families included in the QKP-TIG benchmark. The goal is to provide diverse structural properties that stress-test algorithms under geometric, stochastic, sparse, dense, and application-oriented interaction patterns.
Across all families, item weights are drawn independently (either $w_i \sim \mathrm{UD}[1, 100]$ or $w_i \sim \mathrm{UD}[1, 50]$, depending on the collection) and knapsack capacities are specified through budget fractions .

3.1 Standard QKP

The `Large-QKP` collection follows the classical uniform random generation methodology of Gallo et al. [1], scaled to modern large-scale dimensions.
For each problem size

$$n \in \{500, 1000, 2000, 5000, 10000\},$$

the quadratic interaction matrix is drawn i.i.d. from:

$$p_{ij} \sim \mathrm{UD}\{1, \ldots, 100\},$$

then symmetrised. A density parameter

$$d \in \{5, 10, 15, 20, 25, 50, 75, 100\}\%$$

is applied via independent Bernoulli sparsification of entries.
Weights follow:

$$w_i \sim \mathrm{UD}[1, 50].$$

This collection matches the structure of widely used academic benchmarks while extending them to very large dimensions.

3.2 Dispersion-QKP Collection

The `Dispersion-QKP` collection contains four structurally distinct classes of quadratic profit matrices, each combined with multiple sparsification levels. For each class, a complete $n \times n$ utility matrix is generated and then sparsified by retaining each entry independently with probability $\alpha \in \{0.05, 0.10, 0.25, 0.50, 0.75, 1.00\}$.

1. Geometric (`geo`) Instances

Items correspond to random points uniformly distributed in a 100×100 square. Quadratic profits are defined by Euclidean distances:

$$p_{ij} = \|x_i - x_j\|_2.$$

These instances exhibit strong spatial correlation and metric structure. Larger distances correspond to larger interactions.

2. Weighted Geometric (`wgeo`) Instances

Items again lie uniformly in a square, but each item receives an additional continuous weight factor $u_i \sim \mathrm{UD}[5, 10]$. Interaction profits become:

$$p_{ij} = u_i u_j \|x_i - x_j\|_2.$$

This produces heavy-tailed interaction distributions and amplifies long-range influences.

3. Exponential (`expo`) Instances

Interaction values are drawn from an exponential distribution:

$$p_{ij} \sim \mathrm{Exponential}(\lambda),$$

symmetrised and with zero diagonals. These instances exhibit strong stochastic heterogeneity, with many small interactions and occasional large outliers.

4. Random Uniform (`ran`) Instances

Interaction values are independent integers:

$$p_{ij} \sim \mathrm{UD}\{1, \dots, 100\},$$

followed by symmetrisation. These are fully unstructured and represent classical random-noise QKP test cases.
Each of the above is evaluated at problem sizes $n \in \{300, 500, 1000, 2000\}$.

3.3 TeamFormation-QKP-2 Collection

The `TeamFormation-QKP-2` family models realistic pairwise compatibility derived from project co-membership. It is generated through four steps:

1. A universe of 30,000 projects is partitioned into subsets whose sizes follow a lognormal distribution.
2. Each participant i is assigned a random number of projects (lognormally distributed).
3. Participants draw projects either from a single subset or, when needed, from the full project universe.
4. Pairwise interaction profits are the Jaccard similarities:

$$p_{ij} = \frac{|P_i \cap P_j|}{|P_i| + |P_j| - |P_i \cap P_j|}.$$

These instances produce real-world-like sparse similarity graphs with highly clustered structure. Sizes range from $n = 1000$ up to $n = 10000$.

4 Two-phase Verification: Baseline Calculation

Since all instances are randomly generated client-side, the optimal solution is unknown. To evaluate solution quality and protect against malicious behaviour, a two-phase verification procedure is used.

4.1 Two-tier Verification

Tier 1 — Proof-of-work Baseline

A fast greedy heuristic verifies that submitted solutions meet a minimum quality threshold. This baseline:

1. computes an interaction-aware profit-to-weight ratio for each item,
2. sorts items by this ratio,
3. selects items greedily while respecting capacity.

This tier acts as a lightweight validator to prevent low-effort or adversarial submissions.

Tier 2 — Quality Measurement Baseline

A more advanced two-stage algorithm provides a stable reference performance level against which submitted solutions are scored.

Stage 1: Greedy Construction An initial feasible solution is created by sorting items using an interaction-aware profit-to-weight ratio and selecting them while capacity allows.

Stage 2: Local Search with Tabu List A local search improves the initial solution:

- interaction contributions of each item are precomputed for efficient evaluation;
- add/drop and swap moves are explored to find improving neighbours;
- a tabu list (length 3) prevents cycling;
- moves with insufficient marginal value are discarded early.

5 Challenge Tracks

Within each challenge, there are various challenges tracks. These can range over instance size and/or type. Currently, the challenge supports varying sizes of standard QKP instances:

- 300
- 1000
- 2000
- 5000
- 6000

6 Quality

Solution quality is measured using the *better than baseline* score. Let $\mathrm{s}_{\mathrm{base}}$ denote the Tier 2 baseline objective value and $\mathrm{s}_{\mathrm{alg}}$ the submitted solution's value. The quality score is:

$$\text{better\_than\_baseline} = \frac{s_{\text{alg}} - s_{\text{base}}}{s_{\text{base}}}.$$

Higher scores correspond to stronger performance relative to the sophisticated baseline and yield greater rewards. This incentivises meaningful algorithmic innovation and consistent performance improvements.

References

[1] Giorgio Gallo, Peter L Hammer, and Bruno Simeone. "Quadratic knapsack problems". In: *Combinatorial optimization* (1980), pp. 132–149.

[2] David Pisinger. "The quadratic knapsack problem—a survey". In: *Discrete applied mathematics* 155.5 (2007), pp. 623–648.

[3] Franklin Djeumou Fomeni, Konstantinos Kaparis, and Adam N Letchford. "A cut-and-branch algorithm for the quadratic knapsack problem". In: *Discrete Optimization* 44 (2022), p. 100579.

[4] ME Fennich, LC Coelho, and F Djeumou Fomeni. "Tight upper and lower bounds for the quadratic knapsack problem through binary decision diagram". In: *Les Cahiers du GERAD ISSN* 711 (2024), p. 2440.

[5] Joachim Schauer. "Asymptotic behavior of the quadratic knapsack problem". In: *European Journal of Operational Research* 255.2 (2016), pp. 357–363. DOI: 10.1016/j.ejor.2016.06.01. URL: https://ideas.repec.org/a/eee/ejores/v255y2016i2p357-363.html.

[6] Dorit Hochbaum et al. *A Fast and Effective Breakpoints Algorithm for the Quadratic Knapsack Problem.* Aug. 2024. DOI: 10.48550/arXiv.2408.12183.

[7] Laura Galli, Silvano Martello, and Paolo Toth. "The quadratic knapsack problem". In: *European Journal of Operational Research* (2024). ISSN: 0377-2217. DOI: https://doi.org/10.1016/j.ejor.2024.12.032. URL: https://www.sciencedirect.com/science/article/pii/S0377221724009743.

[8] D.S. Hochbaum et al. "A fast and effective breakpoints heuristic algorithm for the quadratic Knapsack problem". In: *European Journal of Operational Research* (2024). ISSN: 0377-2217. DOI: https://doi.org/10.1016/j.ejor.2024.12.019. URL: https://www.sciencedirect.com/science/article/pii/S0377221724009615.

[9] W David Pisinger, Anders Bo Rasmussen, and Rune Sandvik. "Solution of large quadratic knapsack problems through aggressive reduction". In: *INFORMS Journal on Computing* 19.2 (2007), pp. 280–290.