Instances of the Problem

Instances of the Problem

1. Classical Instances of the Quadratic Knapsack Problem (QKP)

The classical instances of the Quadratic Knapsack Problem (QKP) are generated from a specific Python code. These instances are characterized by:

- **Quadratic and Linear Profits**: A symmetric matrix Q of size $n \times n$, where Q_{ij} represents the profit associated with the simultaneous selection of items i and j. The diagonal elements Q_{ii} correspond to the linear profits (profit from selecting item i alone). The values of Q_{ij} are randomly drawn from $[0, r]$, with a density controlled by a parameter pct% (percentage of non-zero profits).
- **Weights**: Each item i has a weight w_i randomly drawn from $[1, r/2]$.
- **Capacity**: The capacity of the knapsack c is randomly chosen from $[50, \sum w_i - 1]$, where $\sum w_i$ is the total sum of the weights of the items.
- **Objective Function**: Maximize the sum of quadratic and linear profits, i.e., $\max \sum_{i=1}^{n} \sum_{j=1}^{n} Q_{ij} x_i x_j$, where x_i is a binary variable indicating whether item i is selected.
- **Constraint**: The sum of the weights of the selected items must not exceed the capacity of the knapsack, i.e., $\sum_{i=1}^{n} w_i x_i \leq c$.

These instances are often used as benchmarks to test QKP solving algorithms due to their general structure and flexibility (via parameters n, r, and pct%)).

2. Instances of the Dispersion Problem

The instances of the dispersion problem are designed to evaluate the performance of an algorithm on various variants of the problem. Each type of instance is detailed below with its description, objective, characteristics, and an example of application.

2.1 GEO (Geometric Problems)

- **Description**: In this variant, n locations are randomly placed in a square of Hawkins of dimensions 100×100. The distance d_{ij} between two locations i and j is the Euclidean distance, calculated as $d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$, where (x_i, y_i) and (x_j, y_j) are the coordinates of locations i and j.

- **Objective**: Select q locations among the n possible to maximize the sum of the Euclidean distances pairwise between the selected locations, under the constraint $\sum_{i=1}^{n} x_i = q$, where x_i is a binary variable ($x_i = 1$ if location i is selected, $x_i = 0$ otherwise).
- **Characteristics**:
  - The distances depend on the random spatial distribution of the locations, offering a unique geometric configuration per instance.
  - This variant models real scenarios where facilities need to be dispersed in a two-dimensional space (e.g., minimizing signal overlaps).
- **Example of Application**: Placement of telecommunication antennas to avoid signal overlaps.

2.2 WGEO (Weighted Geometric Problems)

- **Description**: Similar to GEO, but each location i receives a weight α_i randomly drawn from $[5, 10]$. The distance d_{ij} becomes $d_{ij} = \alpha_i \alpha_j \times$ Euclidean distance, where the Euclidean distance is defined as in GEO.
- **Objective**: Maximize the sum of the pairwise distances d_{ij} between the q selected locations, under the constraint $\sum_{i=1}^{n} x_i = q$.
- **Characteristics**:
  - The weights α_i amplify the distances, increasing the impact of locations with higher weights.
  - Reflects cases where certain locations have greater importance or capacity.

2.3 EXPO (Exponential Problems)

- **Description**: The distances d_{ij} between each pair $\{i, j\}$ are generated according to a negative exponential distribution with mean 50, with a strong concentration near zero and some large values.
- **Objective**: Maximize the sum of $d_{ij} x_i x_j$ for the q selected locations, under the constraint $\sum_{i=1}^{n} x_i = q$.
- **Characteristics**:
  - The distances are stochastic, without geometric structure.
  - Tests the algorithm's management of irregular data.

2.4 RAN (Random Problems)

- **Description**: Each distance d_{ij} between a pair $\{i, j\}$ is randomly drawn from a uniform distribution in $[1, 100]$.
- **Objective**: Maximize the sum of the pairwise distances d_{ij} between the q selected locations, under the constraint $\sum_{i=1}^{n} x_i = q$.

- **Characteristics**:
  - The distances have no particular structure, serving as a reference case.
  - Evaluates the algorithm's robustness without predefined bias.

2.5 Knapsack-Type Versions (KP-EXPO, KP-GEO, KP-WGEO, KP-RAN)

- **Description**: "Knapsack" variants of the previous types. The constraint $\sum_{i=1}^{n} x_i = q$ is replaced by $\sum_{i=1}^{n} w_i x_i \leq c$, where w_i is a random weight in $[1, 100]$ and $c = \left\lfloor \frac{1}{2} \sum_{i=1}^{n} w_i \right\rfloor$.
- **Objective**: Maximize the sum of $d_{ij} x_i x_j$ while respecting $\sum_{i=1}^{n} w_i x_i \leq c$.
- **Characteristics**:
  - The distances d_{ij} follow the rules of the corresponding versions (EXPO, GEO, WGEO, RAN).
  - Adds a capacity constraint, similar to the quadratic knapsack problem.
  - **Specific Types**:
    * KP-GEO: Euclidean distances with knapsack constraint.
    * KP-WGEO: Weighted distances with knapsack constraint.
    * KP-EXPO: Exponential distances with knapsack constraint.
    * KP-RAN: Uniform distances with knapsack constraint.
- **Example of Application**: Placement of facilities with limited resources (costs or capacities).

2.6 Comparative Tables

Characteristic	**Classical**	**GEO**	**WGEO**
Profits	$Q_{ij} \in [0, r]$, density pct%	Euclidean distances	$\alpha_i \alpha_j d_{ij}$, $\alpha_i \in [5, 10]$
Linear Profits	$Q_{ii} \in [0, r]$	$Q_{ii} = 0$	$Q_{ii} = 0$
Weights	$w_i \in [1, r/2]$	$w_i = 1$	$w_i = 1$
Capacity	$c \in [50, \sum w_i - 1]$	q (cardinality)	q (cardinality)
Objective Function	$\max \sum_{i,j} Q_{ij} x_i x_j$	$\max \sum_{i,j} d_{ij} x_i x_j$	$\max \sum_{i,j} (\alpha_i \alpha_j d_{ij}) x_i x_j$
Constraint	$\sum w_i x_i \leq c$	$\sum x_i = q$	$\sum x_i = q$

Table 1: Classical instances and GEO/WGEO

3. Instances of Densest Subgraphs

3.1 Context and Definition

The densest subgraph problem consists of finding, in a given graph $G = \langle V, E \rangle$, a subset of nodes $U \subseteq V$ of fixed cardinality q, such that the subgraph induced by U contains the largest possible number of edges. In other words, it involves identifying a subset of q nodes that maximizes the density of connections between them.

Characteristic	**Classical**	**EXPO**	**RAN**
Profits	$Q_{ij} \in [0, r]$, density pct%	Exponential distances (mean 50)	Uniform $[1, 10$
Linear Profits	$Q_{ii} \in [0, r]$	$Q_{ii} = 0$	$Q_{ii} = 0$
Weights	$w_i \in [1, r/2]$	$w_i = 1$	$w_i = 1$
Capacity	$c \in [50, \sum w_i - 1]$	q (cardinality)	q (cardinality
Objective Function	$\max \sum_{i,j} Q_{ij} x_i x_j$	$\max \sum_{i,j} d_{ij} x_i x_j$	$\max \sum_{i,j} d_{ij} x_i$
Constraint	$\sum w_i x_i \leq c$	$\sum x_i = q$	$\sum x_i = q$

Table 2: Classical instances and EXPO/RAN

Characteristic	**Classical**	**KP-GEO**	**KP-WGEO**
Profits	$Q_{ij} \in [0, r]$, density pct%	Euclidean distances	$\alpha_i \alpha_j d_{ij}$, $\alpha_i \in [5, 10]$
Linear Profits	$Q_{ii} \in [0, r]$	$Q_{ii} = 0$	$Q_{ii} = 0$
Weights	$w_i \in [1, r/2]$	$w_i \in [1, 100]$	$w_i \in [1, 100]$
Capacity	$c \in [50, \sum w_i - 1]$	$c = \left\lfloor \frac{1}{2} \sum w_i \right\rfloor$	$c = \left\lfloor \frac{1}{2} \sum w_i \right\rfloor$
Objective Function	$\max \sum_{i,j} Q_{ij} x_i x_j$	$\max \sum_{i,j} d_{ij} x_i x_j$	$\max \sum_{i,j} d_{ij} x_i x_j$
Constraint	$\sum w_i x_i \leq c$	$\sum w_i x_i \leq c$	$\sum w_i x_i \leq c$

Table 3: Classical instances and KP-GEO/KP-WGEO

3.2 Formulation as QKP

In the QKP formulation for the densest subgraph problem:

- **Variables**: For each node $i \in V$, a binary variable x_i indicates whether node i is included in the subset U ($x_i = 1$) or not ($x_i = 0$).
- **Objective**: Maximize the number of edges in the induced subgraph, which translates to the following quadratic expression:

  $$\max \sum_{\{i,j\} \in E} x_i x_j$$

  Here, $q_{ij} = 1$ if $\{i, j\}$ is an edge in E, and $q_{ij} = 0$ otherwise. This reflects the fact that the product $x_i x_j = 1$ only if both nodes i and j are included in U and connected by an edge.
- **Constraint**: A cardinality constraint imposes that exactly q nodes are selected:

  $$\sum_{i=1}^{n} x_i = q$$

  In the context of QKP, this constraint is interpreted as a knapsack-type constraint where the weights $w_i = 1$ for all i, and the capacity $c = q$. Since the objective is maximization and the coefficients q_{ij} are non-negative, the optimal solution tends to use the entire capacity, making the constraint effectively an equality.

The instances of densest subgraphs are generated by varying the density of the graph G, i.e., the probability that an edge exists between two nodes. Four types of variants have been defined:

Characteristic	Classical	KP-EXPO	KP-RAN
Profits	$Q_{ij} \in [0, r]$, density pct%	Exponential distances (mean 50)	Uniform $[1, 10$
Linear Profits	$Q_{ii} \in [0, r]$	$Q_{ii} = 0$	$Q_{ii} = 0$
Weights	$w_i \in [1, r/2]$	$w_i \in [1, 100]$	$w_i \in [1, 100]$
Capacity	$c \in [50, \sum w_i - 1]$	$c = \lfloor \frac{1}{2} \sum w_i \rfloor$	$c = \lfloor \frac{1}{2} \sum w_i \rfloor$
Objective Function	$\max \sum_{i,j} Q_{ij} x_i x_j$	$\max \sum_{i,j} d_{ij} x_i x_j$	$\max \sum_{i,j} d_{ij} x_i$
Constraint	$\sum w_i x_i \leq c$	$\sum w_i x_i \leq c$	$\sum w_i x_i \leq c$

Table 4: Classical instances and KP-EXPO/KP-RAN

- **DSUB25**: Each edge is present with a probability of 25%.
- **DSUB50**: Each edge is present with a probability of 50%.
- **DSUB75**: Each edge is present with a probability of 75%.
- **DSUB90**: Each edge is present with a probability of 90%.

For each node i, $d_{ii} = 0$ (no loops), and the parameter q (number of nodes to select) is randomly chosen from the interval $[2, n-2]$, where n is the total number of nodes.

Characteristic	Classical QKP	DSUB
Objective	$\max \sum_{i=1}^{n} \sum_{j=1}^{n} Q_{ij} x_i x_j$	$\max \sum_{\{i,j\} \in E} x_i x_j$
Profit Matrix Q	Symmetric, $Q_{ij} \in [0, r]$	$Q_{ij} = 1$ if $\{i, j\} \in E$, 0 otherwise; $Q_{ii} = 0$
Weights w_i	Random in $[1, \lfloor r/2 \rfloor]$	1 for all i
Capacity c	Random in $[50, \sum_{i=1}^{n} w_i - 1]$	$q \in [2, n-2]$
Constraint	$\sum_{i=1}^{n} w_i x_i \leq c$	$\sum_{i=1}^{n} x_i = q$
Generation	Random profits and weights	Random graphs with probabilities 25%, 50%, 75%

Table 5: Comparison between classical QKP instances and DSUB instances

4. Instances of Planted Cliques in QKP

4.1 Description

The instances of planted cliques are generated as follows:

1. A base graph is created using the Erdős-Rényi model with n nodes, where each possible edge is included with a probability of $\frac{1}{2}$.
2. A clique is planted by selecting k nodes, where

$$k = \lfloor \sqrt{n} \rfloor$$

   (the floor of the square root of n), and then adding edges between all pairs of these nodes, thus forming a complete subgraph.
3. The QKP is configured with:
   - Each node having a weight of 1.

    - The knapsack capacity set to k.
    - The linear profits set to 0 for all nodes.
    - The quadratic profits q_{ij} defined as 1 if an edge exists between nodes i and j, and 0 otherwise.

4. The optimal solution consists of selecting the k nodes of the planted clique, which gives a total profit of $k(k-1)$.

These instances are challenging for QKP algorithms because:

- The base graph is dense due to the edge probability of $\frac{1}{2}$, creating many random connections that obscure the planted clique.

- The planted clique is hidden within this dense graph, making it difficult to distinguish from other dense subgraphs that appear by chance.

- The QKP objective is to maximize the sum of quadratic profits, which amounts to selecting a subset of nodes with the largest possible number of edges between them. The planted clique represents the optimal choice as it has the maximum number of edges, $\binom{k}{2}$, for a subset of size k, leading to a profit of $2\binom{k}{2} = k(k-1)$.

- Identifying the planted clique requires filtering through the noise of the dense graph, which is computationally intensive and tests the algorithm's ability to efficiently find the optimal solution.