The large supply of online products calls for sophisticated techniques to help users explore available items. They propose to build composite items which associate a central item with a set of packages, formed by satellite items, and help users explore them.
First approach
MODEL AND PROBLEM STATEMENT

- Valid and Maximal Packages
- Summarization
- Visual Effect
A valid package must satisfy a given budget such as a visit duration. A maximal package is the largest valid set of satellite items, where each item is compatible with the central item. A valid and maximal package is therefore a set of compatible satellite items, such that, collectively with their central item, satisfy a budget and are not subsumed by another valid package.
Valid and Maximal Packages

- Algorithm Max Composite Item Set: starts from a random single item package and picks the next random item which is different from previously added items and which satisfies compatibility, and validity until the package is maximal.

- Termination Condition:
  - \( P_0 = \frac{n1}{N} \)
  - \( P_0 \) = frequency of all unseen species
  - \( n1 = \) the lone representatives of their species,
  - \( N = \) a random sample
Summarization:

- the goal of summarization is to expose the user to as many satellite items as possible with as few as possible summary packages. Those packages can then be presented to the user, who can directly use them, or select a subset of satellite items to construct their desired composite items, without worrying about checking the budget.
1. Greedy Summarization Algorithms with Bounded Approximation Factors: The algorithm starts by selecting the largest package. At each iteration, it selects the package that, together with the previously chosen packages, produces the highest coverage. The algorithm stops after k packages have been chosen.

2. Randomized Summarization Algorithm: a randomized algorithm, ProbSummarySet, that produces k representative packages directly from the set of compatible satellite items, without generating the full set of maximal packages first.
Visual effect optimization

- there is an ordering of the packages associated with a central item, that minimizes overlap between any two consecutive packages and hence, maximizes their visual diversity
Visual Effect Optimizations

1. Visual Effect Optimizations NP

2. Heuristic Visual Effect Optimization: The basic idea is to always select the next package from among the candidate packages that are optimized for the first satellite type (i.e., the one with the highest priority) and select the package in a greedy fashion by choosing the one that incurs the minimum penalty with the previously chosen package.
Second approach
The composite recommender system consists of one or more recommender systems focusing on different domains. These component RecSys serve top items in non-increasing order of their value (explicit or predicted ratings). In addition, the composite system also has access to information sources (which could be databases or web services) which provide the cost associated with each item.
System Architecture
Problem statement

Given an instance $I$ of a composite recommendation system consisting of one component $\text{RecSys}$ and an external information source, a cost budget $B$ and an integer $k$, find the top-$k$ packages $P_1, \ldots, P_k$ such that each package is feasible and among all feasible packages $P_1, \ldots, P_k$ have the $k$ highest total values, i.e., $v(P) \leq v(P_i)$ for all feasible packages $P \in \{P_1, \ldots, P_k\}$. 
Because of the huge potential size of the sets of items and the high cost of retrieving item information from the source, it is crucial for an algorithm to find high-quality solutions while minimizing the number of items accessed.
Top-1 Composite Recommendation

- First task is to find the sets achieving the maximum value using a recursive function where the total cost is bound by the user’s budget,
- Furthermore, to optimize these sets, using an algorithm to compute the optimal upperbound value of the optimal set.
- Then, the set is exposed to another algorithm to determine the instances with the value bigger than the half of the upperbound optimal value, generating the minimization of the number of items accessed.
Top-k Composite Recommendations

- The same general steps of Top-1 composite recommendations apply here, with a few changes to the algorithms used.
- $k$ is a small constant.
- Enumerating all possible sets using Lawler’s procedure. If the number of sets is at least $k$, then the algorithm reports the top-$k$ packages found; otherwise, it continues accessing the next item.
Greedy algorithm

- A simple greedy heuristic algorithm is used to form a high quality package from *currently accessed items* and then test whether *it* is globally a high quality package.

- Furthermore, instead of using tight upperbound optimal value, an untight heuristic upperbound is used.