- `open_global_cardinality_low_up`,
- `same_and_global_cardinality`,
- `same_and_global_cardinality_low_up`,
- `sum_of_weights_of_distinct_values`,
- `symmetric_cardinality`,
- `symmetric_gcc`,
- `weighted_partial_alldiff`.

A constraint related to assignment problems (i.e., `k_alldifferent`), or a constraint putting a restriction on all items that are assigned to the same equivalence class or on all equivalence classes that are effectively used. Usually an equivalence class corresponds to one single value (see, e.g., the `balance`, `bin_packing`, `global_cardinality`, and `sum_of_weights_of_distinct_values` constraints), to an interval of consecutive values (see, e.g., the `balance_interval`, `interval_and_count`, and `interval_and_sum` constraints) or to all values that are congruent modulo a given number (see, e.g., the `balance_modulo` constraint). The restriction on all items that are assigned to the same equivalence class can for instance be a constraint on the number of items (see, e.g., the `cardinality_atleast`, `cardinality_atmost`, `global_cardinality`, and `global_cardinality_low_up` constraints) or a constraint on the sum of a specific attribute (see, e.g., the `bin_packing`, and `interval_and_sum` constraints).

2.7.15 Assignment dimension

- `coloured_cumulatives`,
- `cumulatives`,
- `diffn`,
- `geost`,
- `geost_time`.

A constraint for handling placement problems involving an assignment dimension. In order to illustrate the notion of *assignment dimension* let us first introduce three typical examples described in Figure 2.2:

- Part (A) of Figure 2.2 considers a scheduling problem where we have both to assign a task to a machine and to fix its start to a time-point, in such a way that two tasks that overlap in time are not assigned to the same machine. In this context the different potential machines where tasks can be assigned is called an assignment dimension. This problem can be directly modelled by a `cumulatives` or a `geost` constraint.

- Part (B) of Figure 2.2 considers a placement problem where we have both to assign a rectangle to a rectangular piece and to locate it within the selected rectangular piece. In this context the different potential rectangular pieces where rectangles can be placed is also called an assignment dimension. Observe that in such placement problems the size of an object in an assignment dimension is always equal to one. This problem can be directly modelled by a `diffn` or a `geost` constraint.

- Part (C) of Figure 2.2 considers a placement problem where we have both to assign a box to a container and to place it within the selected container. In this context the different potential containers where boxes can be packed is also called an assignment dimension. Observe that in such placement problems the size of an object in an assignment dimension is always equal to one. This problem can be directly modelled by a `diffn` or a `geost` constraint.

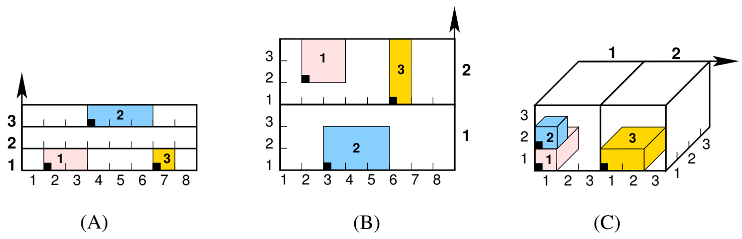

Figure 2.2: Three illustrations of the notion of *assignment dimension* where the assignment dimension is stressed in bold

Using constraints like `coloured_cumulatives`, `cumulatives`, `diffn`, `geost` or `geost_time` allows to model directly with one single global constraint such problems without knowing in advance to which machine, to which rectangular piece, to which container a task, a rectangle, a box will be assigned. For each object the potential values of its assignment variable provide the machines, the rectangular pieces, the containers to which the object can possibly be assigned. Observe that this allows to avoid 0-1 variables for modelling such problems.

2.7.16 At least

- `atleast`,
- `cardinality_atleast`,
- `open_atleast`.

A constraint enforcing that one or several values occur a minimum number of time within a given collection of domain variables.

2.7.17 At most

- `atmost`,
- `cardinality_atmost`,
- `cardinality_atmost_partition`,
- `open_atmost`.

A constraint enforcing that one or several values occur a maximum number of time within a given collection of domain variables.