Scribe Notes: 11/17/2009

Presenter: Chris
Scribe: Robert

Paper: Optimal Rectangle Packing: A Meta-CSP Approach
Authors: Michael D. Moffitt and Martha E. Pollack, AAAI 2006

The speaker stated that the paper

1. Addresses an optimization and a decision problem
2. Describes the techniques for solving them, without providing the formal algorithms
3. Describes the application of general techniques in the context rectangle packing
4. Discusses the design of domain-dependent techniques for the particular application
5. and finally, presents experiments comparing the effectiveness of the techniques proposed to previous ones in the literature.

The Problem

The problem is about arranging (packing) a set of rectangles on a two dimensional surface such as no two rectangles overlap. Two problems are considered:

1. The optimization problem aims at finding the ***minimal bounding box***. Slack denotes the empty space between rectangles. Ideally, the slack should be null.
2. The decision (a.k.a. satisfaction) problem aims at solving the ***containment problem***, i.e. packing the given rectangles in a given *enclosing space*. All rectangles must be contained within the enclosing space.

Typically, any optimization problem can be transformed into a decision problem as follows. Consider an optimization problem that aims to minimize the value of a given objective function. We transform the optimization problem into a satisfaction one by fixing the value of the objective function to a given constant and answering the question: 'does the problem with the given constant have a solution?' In our context, we morph the minimal bounding box problem into a containment problem by choosing a given bounding box and trying to place the rectangles in it.

- If we can solve the problem, then we reduce the size of the bounding box and try to solve the new containment problem.
- If we cannot solve the problem, then we increase the size of the bounding box and try to solve the new containment problem.

The process is repeated until we solve the problem.

Constraint Model of the Containment Problem

The CSP model is defined as follows:

- Variables: Each rectangle i yields two CSP variables, x_i and y_i, corresponding to the horizontal and vertical positions of the rectangle in the enclosing space.
- Domains: The domains of the variables are the integers. Chris noted that it would make more sense to restrict the domains to natural numbers.
- The constraints are of two types: containment constraints and non-overlap constraints
  - Containment constraints force each rectangle to stay within the enclosing space. Those are unary constraints and can in fact be used directly to filter the domains.

- Non-overlap constraints apply to every pair of rectangles and ensure that they do not overlap. Each non-overlap constraint consists of two disjuncts and each disjunct is a linear inequality between two variables each pertaining to a rectangle:
  - The two horizontal disjuncts states that two rectangles are such that one is to the left (L) or to the right (R) of the other.
  - The two vertical disjuncts state that two rectangles are such that one is above (A) or below (B) the other.

➔ Dr. Choueiry noted that each non-overlap constraint is a *disjunction* of four linear inequalities: at least one of them needs to hold for the constraint to be satisfied. It can be R, L, A, or B, but it can also be (R and A) or (R and B), but of course not (R and L).

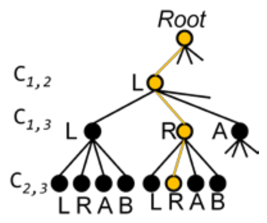
➔ Robert noted that a non-overlap constraint is not "truly" a binary constraint, because its scope is two horizontal and two vertical positions of the two rectangles, though each inequality is a binary constraint and applies to two of the variables at a time.

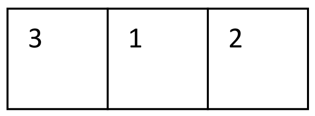
To solve the CSP, the authors propose to model it as meta-CSP:

- Variables: Each constraint becomes a variable of the meta-CSP
- Domains: The domain of each variable is the set of four linear inequalities {R,L,A,B}
- There is only one global constraint in the meta-CSP: it dictates that the inequalities chosen for meta-CSP variables must be such that they form a consistent set of linear inequalities.

A solution to the meta-CSP requires selecting one linear inequality from each meta-variable such that the system of inequalities is consistent.

The strength of the meta-CSP representation is that it does not commit to the 'exact' position of the rectangles in the enclosing box. It finds a 'floating' but consistent arrangement of the rectangles where the positions of the rectangles 'relatively the one to another' are specified: several possible instantiations in terms of 'absolute' positions in the enclosing box are possible.





Like for the TCSP and the DTP, the meta-CSP can be solved with search. The search space is depicted above. $C_{1,2}$ means 1 is to the left of 2, $C_{1,3}$ means 1 is to the right of 3, $C_{2,3}$ means 2 is to the right of 3.

➔ Wesley commented that, if the domain size is 4, then the size of the search space is $4^{\frac{N^2-N}{2}}$, where N is the number of rectangles given as input.

Applying Floyd-Warshall can determine consistency (and minimality) of the system of linear inequalities using the distance graph (similar to the STP). Chris showed how making an assignment (for instance, $C_{1,2} \leftarrow R$) is reflected by adding a weighted, directed edge (between x_1 and x_2) in the distance graph. He also noted that we have two distance graphs (one for the vertical and one for the horizontal constraints) because the horizontal and vertical dimensions do not 'mix' in the inequalities {L,R,A,B}.

Application of 'Traditional' Techniques

They authors describe the application of three 'traditional' techniques:

1. The first technique is *Forward Checking*. After each assignment of a meta-CSP variable, a value of a future variable is checked for consistency with the current path, and removed if it is not consistent.
2. The second technique is the *removal of subsumed variables*. These appear when an implied value can be made explicit. For example, after the instantiation of two meta-CSP variables, the propagation on the distance graph reveals that an unassigned meta-CSP variable must have a particular value. This last unassigned variable is said to be 'subsumed' by the previous two.
   - ➔ One wonders how that really differs from identifying a meta-CSP variable with a 'singleton' domain after forward-checking, as done by the 'domino effect.'
3. The third technique is *semantic branching*. When instantiating a meta-CSP variable fails, the inequality 'assigned' can be negated in that branch, and the algebraic equation resulting from the negation can be added as an additional edge to the distance graph, thus further tightening the distance graph and speeding up propagation.
   - ➔ Dr. Choueiry noted that you need to keep track of subsumed variables / semantic branching for the case backtracking occurs during search, which is not discussed in the paper.

Domain-Specific Techniques

Chris related the use of three domain-specific techniques to improve the performance of search:

1. *Dynamic Symmetry Breaking*. Every constraint has two vertical and two horizontal disjuncts. Due to symmetry, only one of each two disjuncts needs to be considered. If one disjunct succeeds/fails, the symmetric other will also succeed/fail.
2. *Detecting Cliques of Displacement*. Those cliques are used to track/detect when the sum of the widths/heights of the rectangles exceeds the width/height of the enclosing space. Clearly, every re-arrangement of these rectangles will fail. This situation can be detected using our new friend, the displacement graph!
   - o Two such graphs are maintained: vertical and horizontal. Both graphs have a vertex for each rectangle.
     - ▪ In the vertical graph, the nodes are labeled with the height of the rectangle, and
     - ▪ In the horizontal graph the nodes are labeled with the width of the rectangle.
   - o Prior to search and assignments, you add edges to the graph. You add an edge if there is a removal of both of the horizontal or vertical disjuncts. If the vertical disjuncts are removed an edge is added to the horizontal graph (similarly for the horizontal disjuncts). (So, if {L,R} is removed from $C_{2,3}$, there would be an edge between 2 and 3 on the vertical graph). After making the assignment you add the corresponding edge to the appropriate graph.
   - o To get the clique you do two passes, once up and once down.
     - ▪ On the down pass you add the first node to a set of clique nodes then add every vertex to the clique set if it is adjacent to all the vertexes already in the clique set.
     - ▪ On the up pass you find the first node with an edge and add it to the clique set. You then proceed upwards as you did with the downwards pass.
   - o You can then look at the clique and look at the numbers on the vertical and horizontal cliques. The vertical clique tells you the horizontal bounds, and vice-versa.
3. *Variable and value ordering*:

  - Variable ordering: select the largest rectangle (Exploiting the domino effect),
  - Value ordering: choose the one that will cause the minimal increase in area.

Optimization Problem

To solve the optimization problem, the authors use a branch and bound search.

- You start with a partial solution (for example all the rectangles lined up vertically), and record its area.
- You keep track of the best solution's area, A, and the lower bound on the width, w_L, and height, h_L, of the current state. Why the lower bound? Because we just specify relative positions (above, below, left, right) to other rectangles. We never specify how far.
- We want to minimize the area of this state by placing the rectangles as close to each other as possible. In order to determine the lower bound on the width and height, you examine the displacement graphs.
- You only want to continue down the current search path if the current state will fit in an enclosing space with the dimensions W=floor(A/h_L) and H=floor(A/w_L). If these constraints are broken you can backtrack.

Results

Chris discussed the results reported in the paper: both comparing to other techniques and comparing the relative merits of each technique proposed by the authors. While all techniques proposed by the authors seem effective, semantic branching seems to be important, followed by symmetric breaking.

Discussion

- Dr. Choueiry mentioned that, according to the authors, Korf works on absolute positions, not relative. And that Korf is the best teacher EVER! If you are at a conference and Korf is giving a talk, you should make every effort to attend his talk and watch him smoothly conveying to you the most difficult concepts.
- Shant mentioned that all the numerical results seems to be within a constant factor and may well be the result of a good/bad implementation. Everything depends on a malloc!
- It was also discussed that the authors clarified that their approach currently cannot handle rotating rectangles to find better solutions.