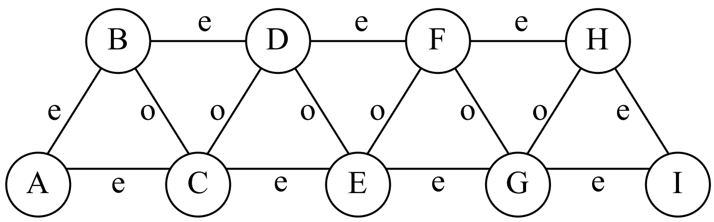
Constraint Propagation Problems

Problem 1

Consider the following constraint network, where all variables have domain {1,2,3}, except variables A and I, whose domain is {0,1,2,3}. The binary constraints labelled with e (resp. o) are satisfied if one of the variables take value 0 or the sum of the variables is even (resp. odd).

.



a) What would be the pruning of the variables domains if node-consistency is maintained? And arc-consistency? Justify.

b) Show that path-consistency would be able to fix the value of some variables? Which ones?

c) Justify whether maintaining arc-consistency would be sufficient to obtain solutions of the problem without backtracking. And path consistency?

Proposed Solution:

a) Maintaining node-consistency only affects unary constraints. As there are none, no pruning is achieved.

For every variable (node) all and each of its neighbours have values with the same and different parity. Hence every value of this variable has some support in all its neighbours, so again no pruning is achieved.

b) Considering path consistency, take the pair of variables <A, B>. Because C has the same parity of A and different parity from B, this means that A and B should have different parities. But given the binary constraint between A and B, these variables either have values with the same parity, or A = 0, hence it must be A = 0.

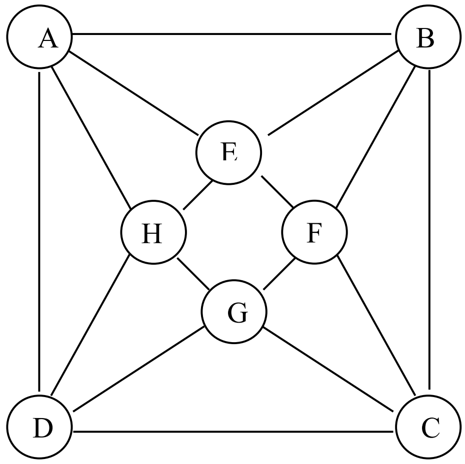
A similar reasoning would fix the value of variable I to I = 0.

c) The ordering A, B, C, … I induces a width of 2 in the graph (every node has at most 2 neighbours with lower ranking. Hence if variables are labelled in this order, no backtracking occurs. As seen above, A is fixed to 0. Then, given constraint A-B, B might take any value v. Then propagation of the other constraints will enforce the other variables to keep only values with compatible parities, i.e. D, F and H will keep values with the same parity of v and C, e and G will keep values with parity different from v, whereas I is already fixed to 0.

Considering arc-consistency, that has not fixed A and I to 0, labelling A with a value different from 0 will cause backtracking since propagation of constraints A-B and A-C will enforce the values remaining in B and C to have the same parity of the chosen value for A, but this is contradictory with the constraint B-C.

Problem 2

Consider the following constraint networks where variables A, B, C and D have domain {1,2,3} and variables E, F, G and H have domain {2,3,4} The binary constraints shown are all difference constraints (≠).



a) Show that the constraint network is node- and arc-consistent. Justify.

b) Assume arc-consistency is maintained on the constraint network. What would be the result of propagation once A is set to 2?

c) Assume now that variables A and B are restricted to the domain {2,3} (and the others keep their previous domains. Show that the problem becomes impossible.

d) Do you think this impossibility would be obtained, without backtracking, by arc-consistency? And path consistency? Justify.

Proposed Solution:

a) Maintaining node-consistency only affects unary constraints. As there are none, no pruning is achieved.

   All variables have at least two values in their domain. Hence, every value of a variable has at least one support in any of its neighbouring variables, so no pruning is achieved by maintaining arc-consistency.

b) Once A is set to 2, maintaining node-consistency, would prune the domains of variables B and D to {1,3} and the domains of variables H and E to {3,4}. Since all variables, except A, would have at least two values in their domain, no further pruning would be achieved.

c) If variables A and B have domain {2,3}, and they are different, variable E should be fixed to 4, since this is the only value in its domain which is different from 2 and 3. Hence variables H and F will have their domain pruned to {2,3}. Hence variables A and H have domain {2,3} which fixes the value for D to D = 1. Similarly, since B and F have domain {2,3} the value for C is fixed to C = 1. But C and D must be different so the problem is impossible.

d) Propagation of constraints of difference to maintain arc-consistency only prunes domains if the domain of one variable is fixed. Since no variables are initially fixed, no pruning is achieved and the network is arc-consistent (but not satisfiable).

   Consider pair A-E with domains A in {2,3} and E in {2,3,4}. Any partial label with this pair, can only be extended to variable B, with domain {2,3} if variable E takes value E = 4, thus fixing its value. Hence variables H and F have their domains pruned to {2,3}. Now, any partial label concerning variables H and A could only be extended to D if D is fixed to D = 1. Similarly, any partial label regarding variables F and B could only be extended to C if it is fixed to D = 1. Fixing the values of C and D to 1 contradicts the constraint C ≠ D, and hence the impossibility of the problem is detected by maintaining path-consistency.

Problem 3

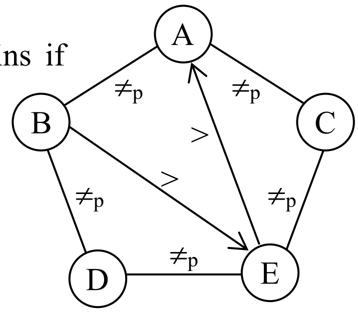
Consider the following constraint network, where all variables have domain {1,2,3,4}. The binary constraints labelled with ≠p are satisfied if the constrained variables have diffeernt parity (i.e. one is even and the other odd). The binary constraints represented by directed arcs A → B should be read as X > Y. For the following questions provide the answers and the adequate justifications.

a) Is the constraint network satisfiable?

b) What would be the pruning of the variable domains if bounds-consistency is maintained?

c) And arc-consistency?

d) And path-consistency?



Proposed Solution:

a) Checking the parity constraints on the variables we notice that A should have different parity of B and C. Hence, both D and E should have the same parity of variable A. But this is impossible, as D and E should have different parity. Hence, the problem is impossible.

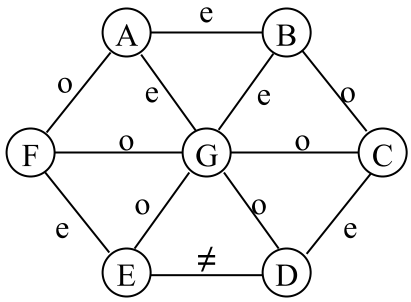
b) Maintaining arc-consistency on the inequality constraints (B>E and E>A) prunes the domais of variables A, E and B to respectively {1,2}, {2,3} and {3,4}. Although reduced, these domains keep values of both even and odd parities. Hence the upper bound of all variables (be it, 2, 3 or 4) is supported by some value of each and all of the neighbouring variables. And so does the lower bound (be it, 1, 2 or 3), leading to no reduction of the domains of variables C and D.

c) Maintenance of arc-consistency leads to the same pruning of bounds-consistency for this problem. The reasoning above regarding the upper- and lower-bound values could be extended for values 2 and 3 of variables C and D (that are not bounds of the variables) but have support in each of the neighbouring variables.

d) Considering labels of the variable pair [A , D], we notice that to be consistently extended to variable B, they must have the same parity. Hence there is an implicit constraint A $=_p$ D that is elicited by maintaining path consistency. Similarly, another implicit constraint, A $=_p$ E, is elicited. Hence, any consistent label regarding variables D and E, that must have values with different parities, may be extended to variable A. Hence, maintaining path-consistency is sufficient to prove that the problem is impossible.

Problem 4

Consider the following constraint network, where nodes correspond to variables and edges to binary constraints. Edges labelled with **e** and **o** denote, respectively, constraints imposing that the sum of the variables is even and odd. Edges labelled with a ≠ correspond to the usual difference constraint. All variables have domain {1,2,3}.



a) What is the domain pruning achieved when node-, arc- and path-consistency is maintained? Justify.

b) Consider now that the constraint between variables A and B becomes of type "o". Verify that the problem becomes unsatisfiable and show what type of consistency should be maintained to detect such unsatisfiability without labelling the variables.

c) For a generic constraint network with the above topology, where enumeration was needed, what variable ordering would you use to do such enumeration? How many variables should be labelled, to reach a backtracking free labelling of the remaining network? Justify.

Proposed Solution:

a) Maintaining node-consistency only affects unary constraints. As there are none, no pruning is achieved.

   All variables have at least two values in their domain and with different parity. Hence, every value of a variable has at least one value that supports it in any of its neighbouring variables, regardless of the binary constraint (e, o or ≠). Hence no pruning is achieved by maintaining arc-consistency.

   To be extended consistently to variable G, any label for the pair of variables D and E should enforce values with the same parity for these variables (D and E). But since they must be different, the only possible pairs for the variables are <D=1, E=3> or <D=3, E=1>, and value 2 is thus pruned from the domain of both variables, which also fixes G to value 2. Further propagation will prune the domains of C and F to {1,3}, and the domains of A and B to the singleton {2}.

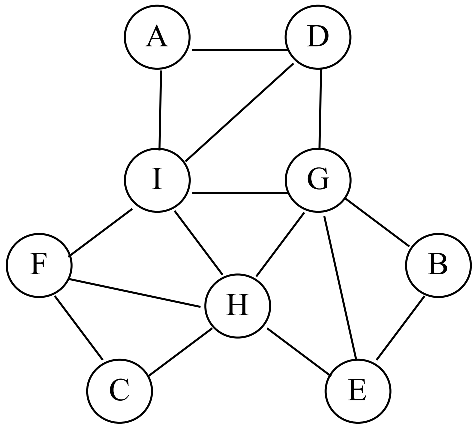
b) For the reason explained above, maintenance of arc-consistency would not prune any domain, since all domains remain with at least two values of different parity.

   Now, maintaining of path-consistent would guarantee that any consistent labelling of the variable pair [A,B] could be extended to variable G. But any consistent label of variables A and B enforces A and B to have values with different parity (e.g. A=1 and B = 2). But then no value for G has different parity from the values chosen for A and B, thus pruning all values from the domains of A and B, thus detecting the unsatisfiability of the problem.

c) In general, a good static variable selection heuristics should adopt some ordering that induces minimal with to the network graph. In this case, this recommends starting with variable G, so that all other variables have at most two neighbouring variables with lower rank. Once G and another variable are labelled, the remaining graph becomes a tree (in fact a sequence) and any labelling becomes backtracking free.

Problem 5

Consider the following constraint network, where all variables have domain {1,2,3} and the binary constraints are constraints of difference (≠).



a) Show that no type of consistency (node-, arc- or path-) may prune the domains of the variables. Suggestion: analyse the solutions of the problem.

b) Assume that variables are labelled according to the ordering [A, B, C, D, E, F, G, H, I]. Show that maintenance of arc-consistency does not guarantee a backtracking free labelling. And path-consistency? Justify your answer.

c) Present an ordering of the variables, static or dynamic, that guarantees a backtracking free search in conjunction with arc consistency. Justify your answer.

d) Could you infer extra-constraints of equality (=) for the problem (keeping equivalence)? Should you make these constraints explicit? Why?

Proposed Solution:

a) Because A and G are variables connected to the same two variables (I and D) that must be different and there are only 3 values in the domains of the variables, there is an implicit constraint that enforces A = G in any solution. Similarly we can obtain the following sets of equal variables: A = F = G, B = H = D, and C = E = I. A can be labelled with an arbitrary value in its domain, so no pruning can be achieved by any type of propagation. Similarly, B and C can take any value provided they are different between them and wrt A, so there are solutions for all values of the variables and hence no value can be pruned from the domain of the variables.

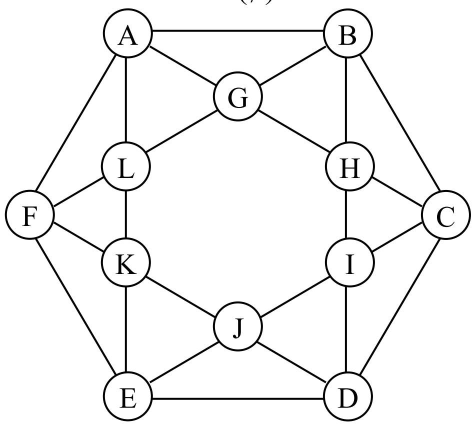
b) From the above A, B and C should all take different values. But if we label A = 1 and B = 1 and C = 1, constraint propagation would prune the domains of all the other variables to {2,3}, and since there are two values in their domains no more pruning would be achieved through propagation of ≠ constraints. Only when variable D is labelled to 2 (or 3) would an inconsistency by detected: on one hand, it should be I = G = 3 (or 2) since they must both be different from D; on the other hand, the constraint I ≠ G prevents variables I and G from taking the same value.

c) Starting the enumeration with the variables with more neighbours (higher degree) is usually a good variable selection heuristics. In this case this suggests starting the labelling with variables G, H and I. As seen before, propagation after labelling G to any of the values in its domain, narrows the domains of H and I to the remaining 2 values. Propagation after labelling H with one of the values, leads to fixing I with the remaining value. From then on, all the values of the other variables will be fixed as discussed above (A = F = G, B = H = D, and C = E = I). Hence a good ordering would start by G, H, I (in any order) followed by the other variables (again in any order).

d) There are implicit ≠ constraints as seen above: A = F = G, B = H = D, and C = E = I. Making them explicit would guarantee any labelling order to be backtracking free: for example, labelling a variable in one of the groups to some value (say A = 1) would fix the values of the other variables of the group (i.e. F = G = 1) to 1, thus narrowing the domain of all the other variables to{2,3}. Then labelling a variable of one of the remaining groups to some value in this narrowed domain (say B = 2) fixes the values of the remaining variables of the group (H = D = 2) and also the variables of the other group (C = E = I = 3).

Problem 6

Consider the following constraint network where all the "external" variables (A to F) have domain {1,2,3} and the others domain {1,2}. All constraints are constraints of difference (≠).



a) Show that the constraint network is not satisfiable.
b) Show that mere maintenance of node- and arc-consistency would not detect such unsatisfiability.
c) And path-consistency?
d) Does any type of consistency achieve domain reduction for any of the variables?
e) Explain what type of consistency and variable heuristics seem more adequate to this network. Will the combination avoid backtracking during search?
f) Show that the problem becomes satisfiable if all variables have initial domain {1,2,3}.
g) How many solutions does the problem have? What additional constraints could be imposed to avoid symmetries? Justify.

Proposed Solution:

a) Variables L and H must take the same value, different from the value of G. So variable A must take the third value, since it must be different from L and from G. Similarly, variable B must take this third value, since it must be different from G and from H. But this is not possible, since A and B must be different.

b) Node-consistency is not applicable, since there are no unary constraints. All variables have at least two values in their domain. Hence, every value of a variable has at least one value that supports it in any of its neighbouring variables (connected by the ≠ constraint). Hence no pruning is achieved by maintaining arc-consistency.

c) Every two variables with two values in the domain (e.g. L and G) are not connected with a third such variable. Hence any two compound labels in these variables can be extended to the third variable (e.g. G and H can be extended to A), so path-consistency will not prune any variable, or impose any extra constraint.

d) From the above items, 5-consistency or higher would achieve such pruning (and detect unsatisfiability). Any label on variables L, G H and I would impose A to take value 3. But it could not be extended to variable B, that should also take value 3, incompatible with that of A.

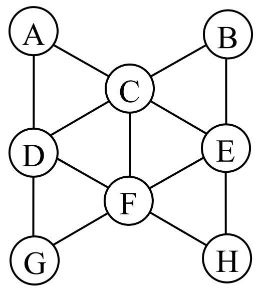
e) After assigning a value to G, arc-consistency would fix the value of all the other inner variables. Moreover. all outer variables, would be connected to two inner variables, so they should have their value fixed to 3. But this is impossible, as connected outer variables must be different. So a good ordering would start with an inner variable, and with 1 backtrack, arc-consistency could detect unsatisfiability.

f) One possible solution would be

A = 1, B = 2, G = 3, H = 1, C = 3, I = 2, D = 1, J = 3, E = 2, K = 1, F = 3, L = 2.

g) Other solutions could be obtained by symmetry with a permutation of the values {1, 2 3}.

Problem 7



Consider the constraint network below, where nodes correspond to variables and arcs to binary constraints.

a) Assuming that all constraints are constraints of difference (≠), show that the network has no solutions if all variables have domain {1,2}. What type of consistency (node-, arc-, path- or higher k- consistency) would allow this conclusion to be inferred, without enumeration of the variables?

b) Now assume that variables C and F have domain {1,3} and all other variables have domain {1,2,3}.

   i. Show that maintaining arc-consistency does not prune the domain of any of the variables.

   ii. In contrast, show that maintaining path-consistency grounds (fixes the value of) some variables. Which variables and what are their values? Are there any other domains pruning?

   iii. Show that the (non-binary) constraint A+B+G+H = 9 is not satisfiable. Would path-consistency on the binary constraints be sufficient to infer this unsatifiability)

c) Assume now that **i.** all binary constraints in the network are arbitrary, **ii.** all variables have *d* values in their domain, and **iii.** arc-consistency is maintained.

   i. Present a static ordering of the variables that should lead to an efficient resolution of the problem. Justify your answer.

   ii. Assuming the same ordering for variable labelling, can you present a weaker type of consistency that, if maintained, would lead to no more backtracks during the labelling of the variables? Justify.

Proposed Solution:

a) Any set of three connected variables (e.g. A, C and D), can be satisfied if the variables have a domain with only two distinct values. Node- and arc-consistency would not detect the issue, but path consistency would detect the inconsistency and the unsatisfiability of the problem.

b)

   i. Since all variables have at least two values in their domain, each of the values has at least one support on a connected variable.

   ii. Any consistent label on variables C and F, can only be extended to variables D and E, by fixing their values to 2. The other variables A, B G and H would also have value 2 removed from their domains. In fact, path-consistency would impose the equality of variables A and B, as well G and H, but would not prune further the domains of these variables.

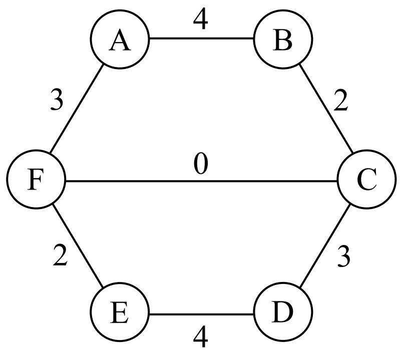
   iii. Because all variables can only take values 1 or 3, and implicitly A=B and G = H, as seen above) the only possible sums would be 4, 8, or 12 (in fact, the sum can only be 8, since variables A and G must be different (applying 4-consistency to variables A, C F and G). Path consistency would not detect the problem, since any compound label on two of the variables A, B, G and H, would leave possible values in the other two variables Only a compound labe in 3 of the variables, say A, B and G could not be extended to the fourth variable (the sum of 3 variables with values 1 and 3 is odd, and to obtain a sum of 9 the remaining variable shoul be even, which is impossible as its domain is {1,3}.

c)

   i. Any order starting with the two variables C and F, would be adequate. After labelling the two variables, the remaining problem is reduced to two independent “trees”.

   ii. As explained in the previous item, node-consistency would be a good trade-off. The only choices to be made (and backtracked) are on variables C and F. The remaining trees are satisfied if they are arc-consistent.

Problem 8



Consider the following constraint network, where all variables have domain {1,2,3,4,5,6}. The binary constraints, labelled with an integer k, are satisfied iff k is the absolute distance between the constrained variables (i.e. if k = 2 and one variable is 3, the other must have values 1 or 5. In particular equality is obtained with k=0). For the following questions provide the answers and the adequate justifications.

a) What would be the pruning of the variable domains if node-consistency is maintained?

b) And bounds-consistency?

c) And arc-consistency?

d) And path-consistency?

e) And (strong) 4-consistency?

Proposed Solution:

a) There are no unary constraints so node-consistency would lead to no pruning of any variable.

b) Bounds consistency would not narrow the bounds of the domains of any variable. In fact, values 1 and 6 of one variable have support in at least on value of the neighbouring variables. For example, A = 1 is supported by F = 5 and B = 5, whereas A = 6 is supported by F = 2 and B = 2.

c) Arc-consistency would narrow the domain of variables A, B, E and D to values {1,2,5,6}. For example, A = 3 does not have a support in B, since B cannot be -1 nor 7. Also, A = 4 has no support in B, since B cannot be 0 nor 8. Hence, the domain of A is pruned from the values 3 and 4, and so is the domain of variables B, E and E.

Now, F cannot be 1 (since A cannot be 4 nor -2), nor 2 (E cannot be 0 nor 4), nor 5 (E cannot be 3 nor 7) nor 6 (E cannot be 4 nor 8). Hence the domain of F is narrowed to {3,4}, and so is the domain of variable C. Hence arc-consistency would narrow the domain of the variables to

A, B, E and D to {1,2,5,6}, and C, F to {3,4}

d) Path consistency would not impose any further constraints, because there are no 3 variables fully connected. For example, labels on C and F (where C = F = 3 or C = F = 4) can be extended to both A and B.

e) Strong 4-consistency would have show the problem is unsatisfiable.In fact any label on C, F and A cannot be extended to B, and any label on C, F and B cannot be extended to A. For example, C = F = 3, impose A = 6, and label {A/6, C/3, F/3} cannot be extended to B, that cannot be simultaneously equal to 2 (because A = 6) and 1 or 5 (because C= 3). The same applies to label {A/1, C/4, F/4}, where B cannot be simultaneously equal to 5 (because A = 1) and 2 or 6 (because C= 4).