Assignment 6
NP Complete Problems
米瑞琪
201718015070043

Problem One Integer Programming

Standard Formulation:

input: an integer matrix with size m*n, a m-dimension vector b.

output: an n-dimension vector x such that Ax>=b.

We are going to prove that Integer programming problem is NPC by reducing 3-SAT to integer programming.

Reduction: For a given 3-SAT problem, we create a integer programming problem as below: Each clause is represented by a formula, e.g. if$C = x_1 \vee x_2 \vee \neg x_3$, the corresponding formula is $x_1 + x_2 + (1 - x_3) \geq 1$, since at least a literal satisfies the clause.

Thus we have a Integer programming problem created from the given 3-sat problem. We now prove that the two problems are equivalent with each other.

Equivalent:

⇨ Suppose we have a true assignment for 3-sat problem , let $x_i = 1$ in integer programming problem if $x_i = TRUE$ in 3-sat problem. Otherwise let $x_i = 0$.

Since each clause is satisfied by at least one literal, we know at each formula can be satisfied.

<= Suppose we have a solution of the integer programming problem , set $x_i = TRUE$ in 3-sat problem if $x_i = 1$ in integer programming problem, otherwise let $x_i = FALSE.$

Problem Two. Mine Sweeper

Standard Formulation:

input: A graph G=(V,E), S⊆V, each node s in S is empty and labeled with a number k representing the number of neighbors with mine. All nodes V in V-S contain mine.

output: “Yes” if such a mine placement is possible; “No” if such a mine placement is impossible.

We can prove that 3 SAT $\leq_p$ mine sweeping to show that mine sweeping problem is a NPC problem. That is, 3-SAT can be mapped into special cases of mine sweeping problem.

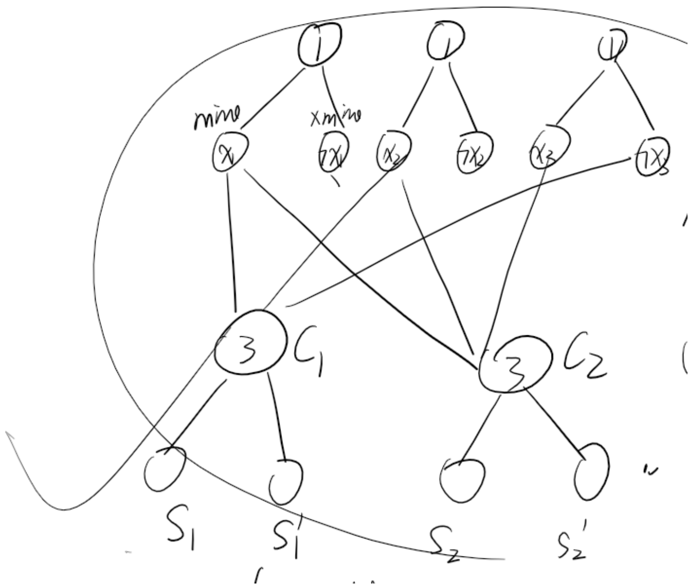
Polynomial Time Reduction:

For a given 3-SAT problem , create a graph G by following strategy:

Suppose there are m clauses in the CNF: $\phi = C_1 \wedge C_2 \wedge \ldots \wedge C_n$, each clause contains 3 literals. e.g. $\phi = (x_1 \vee \neg x_2 \vee \neg x_3) \wedge (x_1 \vee x_2 \vee \neg x_3)$

1) For every variable x_i, set a variable node v_i and two literal node x_i and $\neg x_i$. v_i is labeled with 1.
2) For each clause j, set a clause node C_j, which is labeled with 3.(j=1,2,⋯,m) Connect C_j with its corresponding literal node. For example,$C_1 = (x_1 \vee \neg x_2 \vee \neg x_3)$. We connect C_1 to node $x_1, \neg x_2, \neg x_3$.
3) Each clause node C_j, C_j is connected with 2 extra nodes $s_j\ and\ s_j'$.

.



Equivalence:

=>)

consider a true alignment for CNF ϕ, if $x_i = true$, we put a mine in node x_i and label $\neg x_i = 0$, otherwise we put a mine in $\neg x_i$ label $x_i = 0$.

For each clause node $C_j (j = 1,2, \ldots, m)$:

1 if C_j is satisfied with only 1 literal node, put mine in both of the two extra nodes s_j *and* s_j'.

2 if C_j is connected with 2 literal nodes, put mine in s_j and label $s_j' = 0$.

3 if C_j is connected with 3 literal nodes, label $s_j = 0$ *and* $s_j' = 0$.

Such a graph is a possible mine placement plan since:

1) all variable nodes v_i are labeled with 1, since the assignment is true, only one literal node x_i *or* $\neg x_i$ contains mine and another one is empty(labeled with 0).
2) all clause nodes are connected with 3 mine nodes since it is satisfied by at least one literal, the remaining mine nodes are compensated by s_j or s_j'.

<=)

Since we could determine whether the mine placement for such a graph is possible or not. if the alignment is possible, we say there is a true alignment to 3-SAT problem by set $x_i = true$ if x_i contains a mine, otherwise set $x_i = false$. Since each clause is satisfied by at least one literal and there is no conflict of variable alignment(only x_i $or \neg x_i$ $is\ true\ rather\ than\ both$).

Hence we say Mine sweeper problem is NP Complete.

Problem 3 Half 3-SAT

Formalized Definition:
Input: a CNF with n variables and m clauses(m is even)
Output: Is there an assignment which makes half of clauses are true and another half of clauses are false?
We are going to prove half-3SAT is a NPC problem by reducing 3-SAT to half 3-SAT.
Reduction:
For a given 3-SAT problem with CNF ϕ with n variables and m clauses, we construct a half 3-sat problem as follows: The new half 3-SAT problem contains 4m clauses. we divide the half 3-sat into 3 parts which consists of m, m, 2m clauses separately.
part1. the given 3-SAT problem ϕ(m clauses). Variables in ϕ ranges from x_1 to x_n.
part2. create m clauses which are always true, note that variables are different from variables in ϕ. i.e. $C_{m+1} = x_{n+1} \vee \neg x_{n+1} \vee x_{n+2}, \dots$
part3. create a clause with variables different from part1 and part2, and repeat the clause for 2m times.
The new half 3-SAT problem is of the form :

Part 1. original 3-sat problem with m clauses	Part2. Disjunction of m clauses which are always true(form: $x_{n+1} \vee \neg x_{n+1} \vee x_{n+2}$), the variables in this part is different from part1.	Part3. Repeat an arbitrary clause for 2m times. Variables in this part is different from part 1 and part2 .
Size : m	Size: m	Size:2m

Equivalence:
Denote the original 3-sat problem as ϕ and the corresponding half 3-sat problem as ϕ′, Suppose there is a solution of ϕ′, we claim that ϕ must be satisfiable. Suppose ϕ is not satisfiable, part 2 are always true, no matter part 3 is true or false, the alignment of ϕ′ will never become half true and half false.
Hence we could say that half-3sat is a NPC problem.

Problem 4. Solitaire Problem
Formalized Definition:
input: a n*n board, each of the position lies a blue stone, or red stone, or nothing at all. Remove the stones from the board so that each column contains stones of only one color and each row contains at least one stone.
output: Is the initial configuration a winnable game?

We are going to prove solitaire problem a NPC problem by reducing 3-SAT to Solitaire game. Analyzing the problem we could find that each row can be seen as a clause since there are at least one stone in each row(each clause is at least satisfied by one literal) and each column contains only one color(the value of each variable should not conflict with each other).
Reduction:

for a given 3-sat problem with n variables and m clauses, we can construct a table of solitaire problem as follows:

1) Create m rows , each row represents a clause.
2) Create n columns, each column represents a variable.

Initial configuration: Use red stone to represent TRUE, blue stone to represent FALSE. Suppose literal $C_i = x_{i1} \vee \neg x_{i2} \vee x_{i3}$, put red stone on x_{i1}, x_{i3} and blue stone on $\neg x_{i2}$.

If m=n, the configuration is complete. Otherwise we have to analyze the initial configuration separately:

1) if m<n, which means the number of clauses is less than the number of variables. In this case we add n-m rows and duplicate the alignment of clause C_1 .
2) if m>n, which means the number of clauses is more than the number of variables. In this case we add new clauses from $\mathrm{Cm}+1$,⋯ and make them always true. The variables in the new clauses are different from the variables in the original problem. Whenever we create a new clause(row), we create at least two new variables(column). The growing speed of column is faster than row. Thus by at most (m-n) times of adding we will have a square table.

Equivalence:

Now we have a initial configuration for the game. We can prove that if the game is winnable, there is a true alignment for the 3-sat problem. Since each column have stone with only one color, if the color is red, we set $x_i = TURE$,if the color is blue, we set $x_i = FALSE$. Since each row has at least one stone, the clause can be satisfied by at least one clause.