问题求解作业讲解

EX5-EX6

助教：徐寅

EX5-3.6.1.3

- 直接按照三条定义进行证明即可（该答案引用自陈劭源同学）

Let's verify whether the three conditions in the definition of neighborhood hold. For every input instance Φ of MAX-SAT, define the set of all feasible solutions which differ from x in at most one variable as $f(x)$.

(i) every feasible solution α, since it has no different variable to itself, it belongs to $f(\alpha)$;

(ii) for every feasible solutions α, β, if $\beta \in f(\alpha)$, then β differs from α in at most one variable, and α differs from β in at most one variable too, so $\alpha \in f(\beta)$;

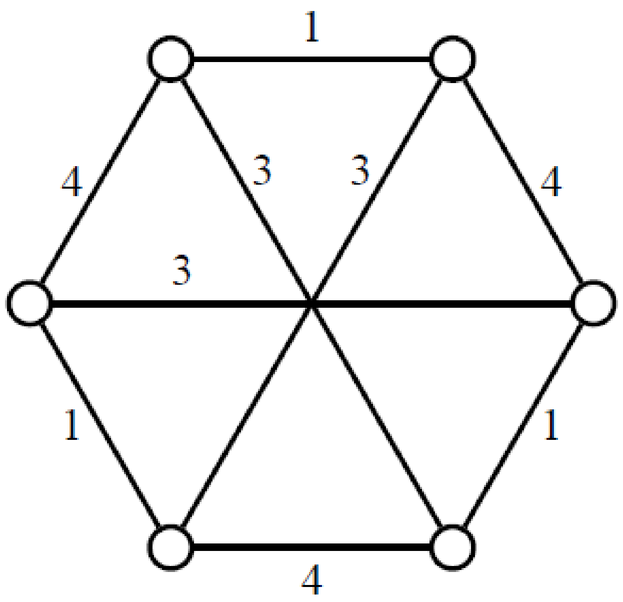
(iii) for every feasible solutions α, β, since they contain finitely many variables, they differ in exactly k variables for some integer k. If $k = 0$, then $\beta = \alpha$, hence $\beta \in f(\alpha)$. Otherwise, define $\lambda_0 = \alpha$, and let λ_i be the feasible solution which is same to λ_{i-1} but the first variable that differs from β is flipped. Hence, $\lambda_k = \beta$, and $\lambda_{i+1} \in f(\lambda_i)$.

Therefore, the transformation described in the problem defined a neighborhood for every input instance Φ of MAX-SAT.

附加题1-1

- 证明解TSP的2-Exchange不是精确的
- 要点：构造反例即可（引用自陈劭源同学）

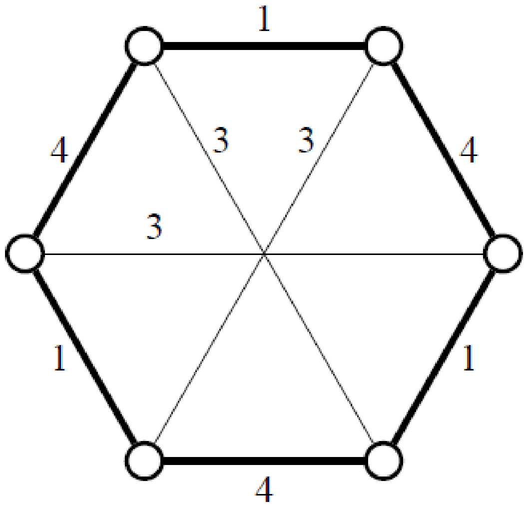
考虑下图的 TSP 问题（所有未画出边的权值为 $+\infty$）

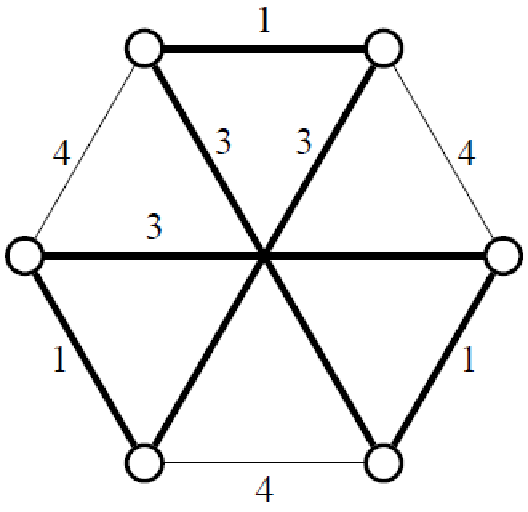


附加题1-1

- 证明解TSP的2-Exchange不是精确的
  - 接上页

下方左图是一个 2-Exchange 邻域下的局部最优解，而右图是一个比左图更优的解。因此 TSP 的 2-Exchange 不是精确的。





附加题1-2

- 证明解TSP的(n-1)-Exchange是精确的
- 题目有些问题，因为(n-1)-Exchange按照定义需要2(n-1)个点，不符合实际，因此看到有些同学把问题转化为n-1次2-Exchange来做，也是可以的。

附加题1-2

- （引用自赵士轩同学）

证：

证明它是精确的，只需证明能够用 $n-1$ 次 2-*Exchange* 将任意一个 Hamiltonian 环变成任意的另一个 Hamiltonian 环。每个 Hamiltonian 环有 n 条边，所以任意两个 Hamiltonian 环最多也就差 n 条边，而 $n-1$ 次 2-*Exchange* 最多能变换 $2n-2$ 条边，我们只需找出一种变换方法即可。根据书中的描述，我们是将两条边的点变为交叉的形式，所以每次会变动两条边。我们只需证每次变动这两条边可以最后使得整个环完成变换。我们设同一个图的任意两个 Hamiltonian 环 $A=\{a_1,\cdots,a_n\}$，$B=\{a_1,b_2,\cdots,b_n\}$（这里我们使得两个环的表述开始于同一个节点）。首先假设 $a_i=b_2$，当 $i\neq n$ 时我们考虑 $\{a_1,a_2\},\{a_i(b_2),a_{i+1}\}$ 这两条边及其四个点，进行一次 2-*Exchange* 操作，有操作后的 $A'=\{a_1,b_2(a_i),a_{i-1},\cdots,a_2,a_{i+1},\cdots,a_n\}$；而当 $i=n$ 时，我们直接重新定义 $A'=a_1,b_2(a_n),\cdots,a_2$（因为 Hamiltonian 环是个环，表述的起始点和方向都可以改！）。我们重复执行这个步骤，直到 $A'=B$。我们会发现进行该操作的次数是 $n-1$ 次，因此我们可以用 $n-1$ 次 2-*Exchange* 将任意一个 Hamiltonian 环变成任意的另一个 Hamiltonian 环，于是原命题得证。 $\square$

附加题2-1

- 要点：根据题目给出的算法中定义出的neighbourhood找到局部最优解的性质（S中加减一个点都不会增加割边权和），从而列出不等式寻找大小关系，（接下来依然引用一下陈劭源同学的证明）

该算法所给的解不会少于最优解的一半。假设 S 是该算法得到的一个局部最优解，则对于 S 中的任意顶点 v，都有

$$\sum_{u\in S} w_{uv} \leq \sum_{u\in V-S} w_{uv}$$

对于 $V-S$ 中的任意顶点 v，也有

$$\sum_{u\in V-S} w_{uv} \leq \sum_{u\in S} w_{uv}$$

否则，在 S 中删去或加入 v 将会获得更优解。

上面的不等式对于所有的 v 求和，分别能得到

$$\sum_{v\in S}\sum_{u\in S} w_{uv} \leq \sum_{v\in S}\sum_{u\in V-S} w_{uv} \tag{1}$$

$$\sum_{v\in V-S}\sum_{u\in V-S} w_{uv} \leq \sum_{v\in V-S}\sum_{u\in V-S} w_{uv} \tag{2}$$

令 C 表示所有割边。将上面的两个式子再相加，然后除以 2（每条边恰好被算了 2 次），可以得到

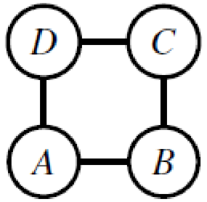
$$\sum_{e\in E-C} w_e \leq \sum_{e\in C} w_e = W(S)$$

令 $W(S_o)$ 表示最优解的边权和。显然，最优解不会超过所有边权和，即

$$W(S_o) \leq \sum_{e\in E} w_e = \sum_{e\in E-C} w_e + \sum_{e\in C} w_e = \sum_{e\in E-C} w_e + W(S)$$

- 然后构造一个例子证明bound是紧的：

如下图所示，下图中每条边的权值为 1，$\{B,C\}$ 是一个局部最优解，割的大小为 2，并且所有只含有 1 个顶点或 3 个顶点的集合构成的割的大小也不会超过 2；然而 $\{A,C\}$ 是一个全局最优解，割的大小为 4。



附加题2-2

- 要点：因为点集的选取可能是指数级的（2^n），所以证明的要点就是构造出一个合数的数据样本，让算法需要遍历指数级的情况。
- 于是这里再分享一下一个有趣的样本构造方法，其设计思路主要有两点：构造出的点集序列需要是大小相差1的；点集序列的割边权和要递增。为了满足这些要求，可以在图中增加很多的散点以纯粹供证明需要。（引用自陈劭源同学）

附加题2-2

例如，考虑图 $G=(V,E)$，其中 $V=\{u_0,u_1,\cdots,u_n,v_0,v_1,\cdots,v_n,t_0,t_1,\cdots,t_n,s\}$, $E=\{u_0v_0,u_1v_1,\cdots,u_nv_n\}$，并且边 u_iv_i 的权值为 2^i。考虑下列点集序列

$$
\begin{aligned}
S_0 &= \{t_0,t_1,t_2,\cdots,t_n\} \\
S_1 &= \{v_0,t_1,t_2,\cdots,t_n,s\} \\
S_2 &= \{t_0,v_1,t_2,\cdots,t_n\} \\
S_3 &= \{v_0,v_1,t_2,\cdots,t_n,s\} \\
&\cdots\cdots \\
S_{2^n-1} &= \{v_0,v_1,v_2,\cdots,v_n,s\}
\end{aligned}
$$

该序列的构造方法为：对于 S_j，若 j 的二进制表示第 k 位（k 从 0 开始）为 0 时，则将 t_k 加入 S_j，否则将 v_k 加入 S_j；此外，如果 j 是奇数，再将 s 加入 S_j。容易验证，S_j 构成割的权值和为 j，并且 $||S_{j+1}|-|S_j||=1$。如果该搜索算法按照上述序列搜索的话，运行时间将达到关于 n 的指数级别。

EX6-3.7.2.1

- 要点：弄清楚standard form，standard inequality form和canonical form格式上的区别
- 等式约束转化为不等式约束：将等式拆成两个不等式，或者在一侧加上一个非负量转化成一个不等式
- 不等式约束转化成等式约束：一般通过增加非负变量将不等式转化为等式

EX6-3.7.2.4

- 严格意义上来说题目有些问题，题目给出的完美匹配的定义和通常我们认为的不一样

A perfect matching is a matching H, where every edge e from E is either in H or shares one common vertex with an edge in H. Express the input instances of this minimization problem as input instances of 0/1-LP.

- 如果按照通常意义上的完美匹配：

Solution:

Minimize $\sum_{e \in E} c(e) \cdot x_e$

under the constraints

$$\sum_{e \in E(v)} x_e = 1 \text{ for every } v \in V$$

$$x_e \in \{0,1\} \text{ for every } e \in E$$

EX6-3.7.2.5

- 要点：优化目标很简单，难点是要保证约束出来的是一棵树
- 方法：根据树的等价条件“连通无环”进行约束（引用自赵士轩同学）

under constraints

$$\sum_{e \in E} x_e = |V| - 1$$

$$\sum_{e \in E, e.u, e.v \in S} x_e \leq |S| - 1 \text{ for every } S \subseteq V$$

and

$$x_e \in \{0, 1\} \text{ for every } e \in E$$

- 当然也有其他的约束方法

EX6-3.7.4.4

- 要点：3.7.4.3的证明中用到的β是可行解，与优化目标无关，因此原证明直接适用。

EX6-3.7.4.12

Solution:

It is easy to show that $Opt_{\text{MMP}}(G) = Opt_{\text{VCP}}(G)$ (König's Theorem) and $Opt_{\text{LP}}\big(I(G)\big) = Opt_{\text{LP}}(Dual(I(G)))$ (LP-Duality Theorem).

Then we show $Opt_{\text{VCP}}(G) = Opt_{\text{LP}}\big(I(G)\big)$.

(1) $Opt_{\text{LP}}\big(I(G)\big) \geq Opt_{\text{VCP}}(G)$

Let $\beta = (\beta_{e_1}, \dots, \beta_{e_n})$ be the optimal solutions of and G.

It is trivial that β is a feasible solution of $I(G)$, so

$$Opt_{\text{LP}}\big(I(G)\big) \geq \sum_{e \in E} \beta_e = Opt_{\text{MMP}}(G) = Opt_{\text{VCP}}(G)$$

(2) $Opt_{\text{VCP}}(G) \geq Opt_{\text{LP}}\big(I(G)\big)$

Let $\alpha = (\alpha_{e_1}, \dots, \alpha_{e_n})$ be the optimal solutions of and G.

Let $S = \{v_1, v_2, \dots, v_k\}$ is the minimum vertex cover and $Opt_{\text{VCP}}(G) = |S| = k$. Since every edge of E is incident to at least one vertex of S,

$$Opt_{\text{LP}}\big(I(G)\big) = \sum_{e \in E} \alpha_e \leq \sum_{v \in S} \sum_{e \in Inc(v)} \alpha_e \leq \sum_{v \in S} 1 = k = Opt_{\text{VCP}}(G)$$

EX6-3.7.4.16

- 要点：构造对偶问题，然后替换约束判断条件

Step 1: Relax *H* to the instance $I_{rel}(H)$ of LP.

$Index(a_j) = \{d \in \{1, \dots, m\} | a_j \in S_d\}$ and $x_i = 1$ iff S_i is picked.

Minimize $\sum_{i=1}^{m} x_i$

Constraints: $\sum_{h \in Index(a_j)} x_h \geq 1$ for $j = 1, \dots, n$; $x_i \geq 0$

Step 2: Construct the dual instance $Dual(I_{rel}(H))$ to the primal instance $I_{rel}(H)$.

$e(S_i) = \{f \in \{1, \dots, n\} | a_f \in S_i\}$ and $y_j = 1$ iff a_j is picked.

Maximize $\sum_{i=1}^{n} y_i$

Constraints: $\sum_{g \in e(S_i)} y_g \leq 1$ for $i = 1, \dots, m$; $y_j \geq 0$

Step 3: Solve $Dual\ (I_{rel}(H))$.

Let $\beta = (\beta_1, \dots, \beta_n)$ be an optimal solution for $Dual\ (I_{rel}(H))$.

Step 4: for $i := 1$ to m do

if $\sum_{j \in Index(a_j)} \beta_j < 1$

then $\alpha_i := 0$

else $\alpha_i := 1$

Output: $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_m)$.

Q&A