组合数学 (Fall 2011)/Optimization

Contents

Duality

Consider the following LP:

$$\begin{align} \text{minimize} \quad & 7x_1 + x_2 + 5x_3 \\ \text{subject to} \quad & x_1 - x_2 + 3x_3 \ge 10 \\ & 5x_1 - 2x_2 - x_3 \ge 6 \\ & x_1, x_2, x_3 \ge 0 \end{align}$$

Let OPT be the value of the optimal solution. We want to estimate the upper and lower bound of OPT.

Since OPT is the minimum over the feasible set, every feasible solution forms an upper bound for OPT. For example $\boldsymbol{x} = (2, 1, 3)$ is a feasible solution, thus $OPT \le 7\cdot 2 + 1 + 5\cdot 3 = 30$.

For the lower bound, the optimal solution must satisfy the two constraints:

$$\begin{align} x_1 - x_2 + 3x_3 &\ge 10, \\ 5x_1 - 2x_2 - x_3 &\ge 6. \end{align}$$

Since the x_i's are restricted to be nonnegative, term-by-term comparison of coefficients shows that

$$7x_1 + x_2 + 5x_3 \ge (x_1 - x_2 + 3x_3) + (5x_1 - 2x_2 - x_3) \ge 16.$$

The idea behind this lower bound process is that we are finding suitable nonnegative multipliers (in the above case the multipliers are all 1s) for the constraints so that when we take their sum, the coefficient of each x_i in the sum is dominated by the coefficient in the objective function. It is important to ensure that the multipliers are nonnegative, so they do not reverse the direction of the constraint inequality.

To find the best lower bound, we need to choose the multipliers in such a way that the sum is as large as possible. Interestingly, the problem of finding the best lower bound can be formulated as another LP:

$$\begin{array}{ll}\text{maximize} & 10y_1+6y_2\\ \text{subject to} & y_1+5y_2\le 7\\ & -y_1+2y_2\le 1\\ & 3y_1-y_2\le 5\\ & y_1,y_2\ge 0\end{array}$$

Here y_1 and y_2 were chosen to be nonnegative multipliers for the first and the second constraint, respectively. We call the first LP the **primal program** and the second LP the **dual program**. By definition, every feasible solution to the dual program gives a lower bound for the primal program.

LP duality

Given an LP in canonical form, called the **primal** LP:

$$\begin{array}{ll}\text{minimize} & \boldsymbol{c}^T\boldsymbol{x}\\ \text{subject to} & A\boldsymbol{x}\ge\boldsymbol{b}\\ & \boldsymbol{x}\ge\boldsymbol{0}\end{array}$$

the **dual** LP is defined as follows:

$$\begin{array}{ll}\text{maximum} & \boldsymbol{b}^T\boldsymbol{y}\\ \text{subject to} & A^T\boldsymbol{y}\le\boldsymbol{c}\\ & \boldsymbol{y}\ge\boldsymbol{0}\end{array}$$

We then give some examples.

Surviving problem (diet problem)

Let us consider the surviving problem. Suppose we have n types of natural food, each containing up to m types of vitamins. The jth food has a_{ij} amount of vitamin i, and the price of the jth food is c_j. We need to consume b_i amount of vitamin i for each $1\le i\le m$ to keep a good health. We want to minimize the total costs of food while keeping healthy. The problem can be formalized as the following LP:

$$\begin{array}{lll}\text{minimize} & c_1x_1+c_2x_2+\cdots+c_nx_n & \\ \text{subject to} & a_{i1}x_1+a_{i2}x_2+\cdots+a_{in}x_n\le b_i & \forall 1\le i\le m\\ & x_j\ge 0 & \forall 1\le j\le n\end{array}$$

The dual LP is

$$\begin{array}{lll}\text{maximize} & b_1y_1+b_2y_2+\cdots+b_ny_m & \\ \text{subject to} & a_{1j}y_1+a_{2j}y_2+\cdots+a_{mj}y_m\le c_j & \forall 1\le j\le n\\ & y_i\ge 0 & \forall 1\le i\le m\end{array}$$

The problem can be interpreted as follows: A food company produces m types of vitamin pills. The company wants to design a pricing system such that

- The vitamin i has a nonnegative price y_i.

- The price system should be competitive to any natural food. A costumer cannot replace the vitamins by any natural food and get a cheaper price, that is, $\sum_{i=1}^{m} y_j a_{ij} \le c_j$ for any $1 \le j \le n$.
- The company wants to find the maximal profit, assuming that the customer only buy exactly the necessary amount of vitamins (b_i for vitamin i).

Maximum flow problem

In the last lecture, we defined the maximum flow problem, whose LP is

$$\begin{align*}
\text{maximize} \quad & \sum_{v:(s,v)\in E} f_{sv} \\
\text{subject to} \quad & f_{uv} \le c_{uv} & \forall (u,v)\in E\\
& \sum_{u:(u,v)\in E} f_{uv} - \sum_{w:(v,w)\in E} f_{vw} = 0 & \forall v\in V\setminus\{s,t\}\\
& f_{uv} \ge 0 & \forall (u,v)\in E
\end{align*}$$

where directed graph $G(V,E)$ is the flow network, $s\in V$ is the source, $t\in V$ is the sink, and c_{uv} is the capacity of directed edge $(u,v)\in E$.

We add a new edge from t to s to E, and let the capacity be $c_{ts}=\infty$. Let E' be the new edge set. The LP for the max-flow problem can be rewritten as:

$$\begin{align*}
\text{maximize} \quad & f_{ts} \\
\text{subject to} \quad & f_{uv} \le c_{uv} & \forall (u,v)\in E\\
& \sum_{u:(u,v)\in E'} f_{uv} - \sum_{w:(v,w)\in E'} f_{vw} \le 0 & \forall v\in V\\
& f_{uv} \ge 0 & \forall (u,v)\in E'
\end{align*}$$

The second set of inequalities seem weaker than the original conservation constraint of flows, however, if this inequality holds at every node, then in fact it must be satisfied with equality at every node, thereby implying the flow conservation.

To obtain the dual program we introduce variables d_{uv} and p_v corresponding to the two types of inequalities in the primal. The dual LP is:

$$\begin{align*}
\text{minimize} \quad & \sum_{(u,v)\in E} c_{uv} d_{uv} \\
\text{subject to} \quad & d_{uv} - p_u + p_v \ge 0 & \forall (u,v)\in E\\
& p_s - p_t \ge 1 \\
& d_{uv} \ge 0 & \forall (u,v)\in E\\
& p_v \ge 0 & \forall v\in V
\end{align*}$$

It is more helpful to consider its integer version:

$$\begin{align}\text{minimize} & \sum_{(u,v)\in E} c_{uv}d_{uv}\\ \text{subject to} & d_{uv}-p_u+p_v\ge 0 & \forall (u,v)\in E\\ & p_s-p_t\ge 1\\ & d_{uv}\in\{0,1\} & \forall (u,v)\in E\\ & p_v\in\{0,1\} & \forall v\in V\end{align}$$

In the last lecture, we know that the LP for max-flow is totally unimordular, so is this dual LP, therefore the optimal solutions to the integer program are the optimal solutions to the LP.

The variables p_v defines a bipartition of vertex set V. Let $S=\{v\in V\mid p_v=1\}$. The complement $\bar{S}=\{v\in V\mid p_v=1\}$.

For 0/1-valued variables, the only way to satisfy $p_s-p_t\ge 1$ is to have $p_s=1$ and $p_t=0$. Therefore, $(S,\bar{S})$ is an s-t cut.

In an optimal solution, $d_{uv}=1$ if and only if $u\in S,v\in\bar{S}$ and $(u,v)\in E$. Therefore, the objective function of an optimal solution $\sum_{u\in S,v\not\in S \atop (u,v)\in E}c_{uv}$ is the capacity of the minimum s-t cut $(S,\bar{S})$.

Duality theorems

Let the primal LP be:

$$\begin{align}\text{minimize} & \boldsymbol{c}^T\boldsymbol{x}\\ \text{subject to} & A\boldsymbol{x}\ge\boldsymbol{b}\\ & \boldsymbol{x}\ge\boldsymbol{0}\end{align}$$

Its dual LP is:

$$\begin{align}\text{maximum} & \boldsymbol{b}^T\boldsymbol{y}\\ \text{subject to} & A^T\boldsymbol{y}\le\boldsymbol{c}\\ & \boldsymbol{y}\ge\boldsymbol{0}\end{align}$$

Theorem

The dual of a dual is the primal.

Proof.

The dual program can be written as the following minimization in canonical form:

$$\begin{align}\min & -\boldsymbol{b}^T\boldsymbol{y}\\ \text{s.t.} & -A^T\boldsymbol{y}\ge-\boldsymbol{c}\\ & \boldsymbol{y}\ge\boldsymbol{0}\end{align}$$

Its dual is:

$$\begin{aligned} \max \quad & -\boldsymbol{c}^T\boldsymbol{x} \\ \text{s.t.} \quad & -A\boldsymbol{x} \le -\boldsymbol{b} \\ & \boldsymbol{x} \ge \boldsymbol{0} \end{aligned}$$

which is equivalent to the primal:

$$\begin{aligned} \min \quad & \boldsymbol{c}^T\boldsymbol{x} \\ \text{s.t.} \quad & A\boldsymbol{x} \ge \boldsymbol{b} \\ & \boldsymbol{x} \ge \boldsymbol{0} \end{aligned}$$

$\square$

We have shown that feasible solutions of a dual program can be used to lower bound the optimum of the primal program. This is formalized by the following important theorem.

Theorem (Weak duality theorem)

If there exists an optimal solution to the primal LP:

$$\begin{aligned} \min \quad & \boldsymbol{c}^T\boldsymbol{x} \\ \text{s.t.} \quad & A\boldsymbol{x} \ge \boldsymbol{b} \\ & \boldsymbol{x} \ge \boldsymbol{0} \end{aligned}$$

then,

$$\begin{aligned} \min \quad & \boldsymbol{c}^T\boldsymbol{x} \\ \text{s.t.} \quad & A\boldsymbol{x} \ge \boldsymbol{b} \\ & \boldsymbol{x} \ge \boldsymbol{0} \end{aligned} \quad \ge \quad \begin{aligned} \max \quad & \boldsymbol{b}^T\boldsymbol{y} \\ \text{s.t.} \quad & A^T\boldsymbol{y} \le \boldsymbol{c} \\ & \boldsymbol{y} \ge \boldsymbol{0} \end{aligned}$$

Proof.

Let $\boldsymbol{x}$ be an arbitrary feasible solution to the primal LP, and $\boldsymbol{y}$ be an arbitrary feasible solution to the dual LP.

We estimate $\boldsymbol{y}^T A\boldsymbol{x}$ in two ways. Recall that $A\boldsymbol{x} \ge \boldsymbol{b}$ and $A^T\boldsymbol{y} \le \boldsymbol{c}$, thus

$$\boldsymbol{y}^T\boldsymbol{b} \le \boldsymbol{y}^T A\boldsymbol{x} \le \boldsymbol{c}^T\boldsymbol{x}.$$

Since this holds for any feasible solutions, it must also hold for the optimal solutions.

$\square$

A harmonically beautiful result is that the optimums of the primal LP and its dual are equal. This is called the strong duality theorem of linear programming.

Theorem (Strong duality theorem)

If there exists an optimal solution to the primal LP:

$$\begin{aligned} \min \quad & \boldsymbol{c}^T\boldsymbol{x} \\ \text{s.t.} \quad & A\boldsymbol{x} \ge \boldsymbol{b} \\ & \boldsymbol{x} \ge \boldsymbol{0} \end{aligned}$$

then,

$$\begin{array}{llcll} \min & \boldsymbol{c}^T\boldsymbol{x} & = & \max & \boldsymbol{b}^T\boldsymbol{y} \\ \text{s.t.} & A\boldsymbol{x} \ge \boldsymbol{b} & & \text{s.t.} & A^T\boldsymbol{y} \le \boldsymbol{c} \\ & \boldsymbol{x} \ge \boldsymbol{0} & & & \boldsymbol{y} \ge \boldsymbol{0} \end{array}$$

Unimodularity

Integer Programming

Consider the **maximum integral flow** problem: given as input a flow network $(G(V,E),c,s,t)$ where for every $uv\in E$ the capacity c_{uv} is integer. We want to find the integral flow $f:E\rightarrow\mathbb{Z}$ with maximum value.

The mathematical programming for the problem is:

$$\begin{array}{lrll} \text{maximize} & \displaystyle\sum_{v:(s,v)\in E} f_{sv} & & \\ \text{subject to} & & f_{uv} \le c_{uv} & \forall (u,v)\in E \\ & \displaystyle\sum_{u:(u,v)\in E} f_{uv} - \sum_{w:(v,w)\in E} & f_{vw} = 0 & \forall v\in V\setminus\{s,t\} \\ & & f_{uv}\in\mathbb{N} & \forall (u,v)\in E \end{array}$$

where $\mathbb{N}$ is the set of all nonnegative integers. Compared to the LP for the max-flow problem, we just replace the last line $f_{uv}\ge 0$ with $f_{uv}\in\mathbb{N}$. The resulting optimization is called an **integer programming (IP)**, or more specific **integer linear programming (ILP)**.

Due to the Flow Integrality Theorem, when capacities are integers, there must be an integral flow whose value is maximum among all flows (integral or not). This means the above IP can be efficiently solved by solving its LP-relaxation. This is usually impossible for general IPs.

Generally, an IP of canonical form is written as

$$\begin{aligned} \text{maximize} \quad & \boldsymbol{c}^T\boldsymbol{x} \\ \text{subject to} \quad & A\boldsymbol{x} \geq \boldsymbol{b} \\ & \boldsymbol{x} \geq \boldsymbol{0} \\ & \boldsymbol{x} \in \mathbb{Z}^n \end{aligned}$$

Consider the **3SAT** problem. Each instance is a **3CNF(conjunctive normal form)**: $\bigwedge_{i=1}^{m}(\ell_{i_1} \vee \ell_{i_2} \vee \ell_{i_3})$, where each $(\ell_{i_1} \vee \ell_{i_2} \vee \ell_{i_3})$ is a **clause** and each $\ell_{i_r} \in \{x_j, \neg x_j \mid 1 \leq j \leq n\}$, called a **literal**, is either a boolean variable or a negation of a boolean variable. We want to determine whether there exists an truth assignment of the n boolean variables $x_1, \ldots, x_n$ such that the input formula is satisfied (i.e., is true).

The following IP solves 3SAT:

$$\begin{aligned} \text{maximize} \quad & \sum_{i=1}^{m} z_i \\ \text{subject to} \quad & z_i \leq y_{i_1} + y_{i_2} + y_{i_3} && \forall 1 \leq i \leq m \\ & y_{i_r} \leq x_j && \text{if } \ell_{i_r} = x_j \\ & y_{i_r} \leq 1 - x_j && \text{if } \ell_{i_r} = \neg x_j \\ & z_i, x_j \in \{0,1\} && \forall 1 \leq i \leq m, 1 \leq j \leq n \end{aligned}$$

Since 3SAT is NP-hard (actually it is the first problem known to be NP-hard), generally IP is NP-hard.

Integrality of polytopes

A point in an n-dimensional space is integral if it belongs to $\mathbb{Z}^n$, i.e., if all its coordinates are integers.

A polyhedron is said to be **integral** if all its vertices are integral.

An easy observation is that an integer programming has the same optimal solutions as its LP-relaxation when the polyhedron defined by the LP-relaxation is integral.

Theorem (Hoffman 1974)

> If a polyhedron P is integral then for all integer vectors $\boldsymbol{c}$ there is an optimal solution to $\max\{\boldsymbol{c}^T\boldsymbol{x} \mid \boldsymbol{x} \in P\}$ which is integral.

Proof.

> There always exists an optimal solution which is a vertex in P. For integral P, all vertices are integral.
>
> $\square$

Unimodularity and total unimodularity

Definition (Unimodularity)

An $n \times n$ integer matrix A is called **unimodular** if $\det(A) = \pm 1$.
An $m \times n$ integer matrix A is called **total unimodular** if every square submatrix B of A has $\det(B) \in \{1, -1, 0\}$, that is, every square, nonsingular submatrix of A is unimodular.

A totally unimodular matrix defines a integral convex polyhedron.

Theorem

Let A be an $m \times n$ integer matrix.
If A is totally unimodualr, then for any integer vector $\boldsymbol{b} \in \mathbb{Z}^n$ the polyhedron $\{\boldsymbol{x} \in \mathbb{R}^n \mid A\boldsymbol{x} = \boldsymbol{b}, \boldsymbol{x} \geq \boldsymbol{0}\}$ is integral.

Proof.

Let B be a basis of A, and let $\boldsymbol{b}'$ be the corresponding coordinates in $\boldsymbol{b}$. A basic solution is formed by $B^{-1}\boldsymbol{b}'$ and zeros. Since A is totally unimodular and B is a basis thus nonsingular, $\det(B) \in \{1, -1, 0\}$. By Cramer's rule (http://en.wikipedia.org/wiki/Cramer's_rule), B^{-1} has integer entries, thus $B^{-1}\boldsymbol{b}'$ is integral. Therefore, any basic solution of $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}, \boldsymbol{x} \geq \boldsymbol{0}$ is integral, which means the polyhedron $\{\boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}, \boldsymbol{x} \geq \boldsymbol{0}\}$ is integral.

$\square$

Our next result is the famous Hoffman-Kruskal theorem.

Theorem (Hoffman-Kruskal 1956)

Let A be an $m \times n$ integer matrix.
If A is totally unimodualr, then for any integer vector $\boldsymbol{b} \in \mathbb{Z}^n$ the polyhedron $\{\boldsymbol{x} \in \mathbb{R}^n \mid A\boldsymbol{x} \geq \boldsymbol{b}, \boldsymbol{x} \geq \boldsymbol{0}\}$ is integral.

Proof.

Let $A' = [\,A \quad -I\,]$. We claim that A' is also totally unimodular. Any square submatrix B of A can be written in the following form after permutation:

$$B = \begin{bmatrix} C & 0 \\ D & I \end{bmatrix}$$

where C is a square submatrix of A and I is identity matrix. Therefore,

$$\det(B) = \det(C) \in \{1, -1, 0\},$$

thus A' is totally unimodular.

Add slack variables to transform the constraints to the standard form $A'\boldsymbol{z}=\boldsymbol{b},\boldsymbol{z}\ge\boldsymbol{0}$. The polyhedron $\{\boldsymbol{x}\mid A\boldsymbol{x}\ge\boldsymbol{b},\boldsymbol{x}\ge\boldsymbol{0}\}$ is integral if the polyhedron $\{\boldsymbol{z}\mid A'\boldsymbol{z}=\boldsymbol{b},\boldsymbol{z}\ge\boldsymbol{0}\}$ is integral, which is implied by the total unimodularity of A'.

$\square$

Matroid

The matroid is a structure shared by a class of optimization problems for which greedy algorithms work.

Kruskal's greedy algorithm for MST

For the **minimum-weight spanning tree (MST)** problem. We are given a connected undirected graph $G(V,E)$ with positive edge weights $w:E\rightarrow\mathbb{R}^+$, and want to find a spanning tree T of the minimum accumulated weight $\sum_{e\in T}w_e$.

We consider the equivalent maximum problem to find a spanning tree with maximum weight. To see this makes the problem no harder, we can replace the weight w_e for every edge $e\in E$ by $W-w_e$, where W is a sufficient large constant which is greater than all weights. The minimum-weight spanning tree for the modified weight is the maximum-weight spanning tree for the original weight.

The following greedy algorithm solves the maximum-weight spanning tree problem.

Kruskal's Algorithm

$S=\emptyset$;
while $\exists e\in E$ that $S\cup\{e\}$ is forest

pick such e with maximum w_e;
$S=S\cup\{e\}$;

It is not hard to verify the correctness of this greedy algorithm. But we are more interested in the general framework underlying this algorithm.

Matroids

Let X be a finite set and $\mathcal{F}\subseteq 2^X$ be a family of subsets of X. A member set $S\in\mathcal{F}$ is called **maximal** if $S\cup\{x\}\not\in\mathcal{F}$ for any $x\in X\setminus S$.

For $Y\subseteq X$, denote $\mathcal{F}_Y=\{S\in\mathcal{F}\mid S\subseteq Y\}$. Obviously,$\mathcal{F}_Y=\mathcal{F}\cap 2^Y$.

Definition

A set system $\mathcal{F}\subseteq 2^X$ is a **matroid** if it satisfies:

- (hereditary) if $T \subseteq S \in \mathcal{F}$ then $T \in \mathcal{F}$;
- (matroid property) for every $Y \subseteq X$, all maximal $S \in \mathcal{F}_Y$ have the same $|S|$.

Suppose $\mathcal{F}$ is a matroid. Some matroid terminologies:

- Each member set $S \in \mathcal{F}$ is called an **independent set**.
- A maximal independent subset of a set $Y \subset X$, i.e., a maximal $S \in \mathcal{F}_Y$, is called a **basis** of Y.
- The size of the maximal $S \in \mathcal{F}_Y$ is called the **rank** of Y, denoted $r(Y)$.

Graph matroids

Let $G(V, E)$ be a graph. Define a set system with ground set E as

$$\mathcal{F} = \{S \subseteq E \mid \textbf{there is no cycle in } S\}.$$

That is, $\mathcal{F}$ is the set of all forests in G.

We claim that $\mathcal{F}$ is a matroid.

First, $\mathcal{F}$ is hereditary since any subgraph of a forest must also be a forest.

We then verify the matroid property of $\mathcal{F}$. Let $Y \subseteq E$ be an arbitrary subgraph of G. Suppose Y has k connected components. For any maximal forest S in Y (i.e., S is a spanning forest in Y), it holds that $|S| = n - k$. In other words, for any $Y \subseteq E$, all maximal member of $\mathcal{F}_Y$ have the same cardinality.

Therefore, $\mathcal{F}$ is a matroid. Each independent set (of matroid) is a forest in G. For any subgraph $Y \subseteq G$, the rank of Y is the size of a spanning forest of Y.

Linear matroids

Let A be an $m \times n$ matrix. Define a set system $\mathcal{F} \subseteq 2^{[n]}$ as

$$\mathcal{F} = \{S \subseteq [n] \mid S \textbf{ is a set of linearly independent columns in } A\}.$$

$\mathcal{F}$ is hereditary since every any subset of a set of linearly independent vectors is still linearly independent.

For any subset $Y \subseteq [n]$ of columns of A. Let B be the submatrix composed by these columns. Then $\mathcal{F}_Y$ contains all sets of linearly independent columns of B. Clearly, all maximal such sets have the same size, which is the column-rank of B.

Therefore, $\mathcal{F}$ is a matroid. Each independent set (of matroid) is a linearly independent set of columns of matrix A. For any set $Y \subseteq [n]$ of columns of matrix A, the rank of Y is the column-rank of the submatrix defined by the columns in Y.

Weighted matroid maximization

Consider the following **weighted matroid maximization** problem. Let $\mathcal{F} \subseteq 2^X$ be a matroid. We define positive weights $w : X \to \mathbb{R}^+$ of elements in X. Our goal is to find an independent set $S \in \mathcal{F}$ with the maximum accumulated weight $\sum_{x \in S} w(x)$.

We then introduce the Greedy Algorithm which finds the maximum-weight independent set.

Greedy Algorithm

$S=\emptyset$;
while $\exists x\not\in S$ with $S\cup\{x\}\in\mathcal{F}$

choose such x with maximum $w(x)$;
$S=S\cup\{x\}$;

The correctness of the greedy algorithm is due to the next theorem.

Theorem (Rado 1957; Edmonds 1970)

The greedy algorithm finds an independent set $S\in\mathcal{F}$ with the maximum weight.

Proof.

Suppose the theorem is false. Let S be the independent set returned by the greedy algorithm and let T be a maximum-weight independent set.

Suppose $S=\{x_1,x_2,\ldots,x_m\}$, where the x_is are chosen by the algorithm in that order. Then it is easy to see that $w(x_1)\ge w(x_2)\ge\cdots\ge w(x_m)$.

Suppose $T=\{y_1,y_2,\ldots,y_\ell\}$, where $w(y_1)\ge w(y_2)\ge\cdots\ge w(y_\ell)$.

Choose the least index k such that $w(x_k)>w(y_k)$. If none exists, then we must have $\ell>m$; in this case we can let $k=m+1$.

In either case we know that the greedy algorithm did not add any of $y_1,\ldots,y_k$ in step k. Since what it did choose has smaller weight, it must be that y_i, for $1\le i\le k$, has the property either that $y_i\in\{x_1,\ldots,x_{k-1}\}$ or that $\{x_1,\ldots,x_{k-1},y_i\}\not\in\mathcal{F}$. In other words, $\{x_1,\ldots,x_{k-1}\}$ is a basis of $Y=\{x_1,\ldots,x_{k-1},y_1,\ldots,y_k\}$. But this contradicts the matroid property, since $\{y_1,\ldots,y_k\}$, being a subset of T, is also an independent subset of Y and is larger.

$\square$