همان طور که در مثال نیز مشاهده شد محاسبه ماتریس $\tilde{A}$ در هر تکرار، هم از لحاظ تعداد محاسبات و هم از لحاظ زمانی مقرون به صرفه نیست. بنابراین تضمین کارایی الگوریتم گفته شده در گرو امکان به کار بردن آن بدون نیاز به محاسبه تک تک درایه های ماتریس $\tilde{A}_i$ می باشد. یک راه حل برای این مشکل به این صورت است که اندیس های $t_i$ و $j_i$ را به گونه ای انتخاب کنیم که $\tilde{r}^i_{t_i}$ و $\tilde{c}^i_{j_i}$ و در نتیجه $(\tilde{r}^i_{t_i} - ۱)(\tilde{c}^i_{j_i} - ۱)$ به صورت قابل انتظاری کوچک باشند. از آنجا که طبق رابطه ؟؟، $t$امین سطر $a_t^T H_i^T$ ،$\tilde{A}_i$ است پس $\tilde{r}^i_t$ تعداد غیر صفرها در $a_t^T H_i^T$ می باشد. بنابراین اندیس سطر $t_i$ از ماتریس $A$ را به گونه ای انتخاب می کنیم که از بین تمام سطرهای ماتریس $A$ که تاکنون در نظر گرفته نشده دارای کمترین درایه غیر صفر باشد و قرار می دهیم $s_i = H_i a_{t_i}$. برای این کار کافی است قبل از شروع الگوریتم سطر های ماتریس $A$ را بر اساس تعداد درایه های غیر صفر به صورت صعودی مرتب کنیم. به طور مشابه از آنجا که $j$ امین ستون ماتریس $\tilde{A}_i$، $A_{m-i}h_j$ است $j_i$ را اندیس ستونی از ماتریس $\tilde{A}_i$ قرار می دهیم که از بین تمام درایه های غیر صفر بردار $s_i$ صادق در نامساوی ؟؟، متناظر با سطری از $H_i$ باشد که دارای کمترین تعداد ناصفر ها است. بنابراین با ترفند گفته شده نیاز به محاسبه ماتریس $\tilde{A}$ در هر مرحله از بین می رود.

ما برای توضیح ایده خود در ابتدا فرض کردیم ماتریس $A$ دارای رتبه سطری کامل باشد. با این وجود در الگوریتمی که در ادامه خواهیم گفت نیازی به این محدودیت نیست (اگر $s_i = ۰$، آنگاه سطر فعلی را رها کرده و سطر بعدی را در نظر خواهیم گرفت). انتظار می رود الگوریتم پیشنهاد شده بتواند یک پایه پوچ تنک برای ماتریس تنک تولید کند. شرح الگوریتم به صورت زیر می باشد.

از آنجا که طبق رابطه ؟؟، $t$امین سطر $a_t^T H_i^T$ ،$\tilde{A}_i$ است پس $\tilde{r}^i_t$ تعداد غیر صفرها در $a_t^T H_i^T$ می باشد. بنابراین اندیس سطر $t_i$ از ماتریس $A$ را به گونه ای انتخاب می کنیم که از بین تمام سطرهای ماتریس $A$ که تاکنون در نظر گرفته نشده دارای کمترین درایه غیر صفر باشد و قرار می دهیم $s_i = H_i a_{t_i}$. برای این کار کافی است قبل از شروع الگوریتم سطر های ماتریس $A$ را بر اساس تعداد درایه های غیر صفر به صورت صعودی مرتب کنیم. به طور مشابه از آنجا که $j$ امین ستون ماتریس $\tilde{A}_i$، $A_{m-i}h_j$ است $j_i$ را اندیس ستونی از ماتریس $\tilde{A}_i$ قرار می دهیم که از بین تمام درایه های غیر صفر بردار $s_i$ صادق در نامساوی ؟؟، متناظر با سطری از $H_i$ باشد که دارای کمترین تعداد ناصفر ها است. بنابراین با ترفند گفته شده نیاز به محاسبه ماتریس $\tilde{A}$ در هر مرحله از بین می رود.

ما برای توضیح ایده خود در ابتدا فرض کردیم ماتریس $A$ دارای رتبه سطری کامل باشد. با این وجود در الگوریتمی که در ادامه خواهیم گفت نیازی به این محدودیت نیست (اگر $s_i = ۰$، آنگاه سطر فعلی را رها کرده و سطر بعدی را در نظر خواهیم گرفت). انتظار می رود الگوریتم پیشنهاد شده بتواند یک پایه پوچ تنک برای ماتریس تنک تولید کند. شرح الگوریتم به صورت زیر می باشد.

**الگوریتم ۱** الگوریتم ...برای محاسبه یک پایه پوچ تنک

**مرحله ۱.** سطرهای ماتریس $A \in R^{m,n}$ را بر اساس تعداد درایه های غیر صفر به صورت صعودی مرتب کرده و آنها را به ترتیب $a_{t_1}$ تا $a_{t_m}$ بنامید. همچنین فرض کنید $H_1$ ماتریس همانی در $R^{n,n}$ و $u \in [0\,1]$ یک عدد ثابت باشد. قرار دهید $r_1 = n$ و $i = 1$.

**مرحله ۲.** قرار دهید $s_i = H_i a_{t_i}$.

**مرحله ۳.** اگر $s_i = 0$ قرار دهید $r_{i+1} = r_i$، $i = i + 1$، $H_{i+1} = H_i$ و به گام دوم بروید.

**مرحله۴.** فرض کنید $s_i^{j_i}$ درایه ی غیر صفری از $s_i$ باشد که از بین همه درایه های غیر صفر $s_i$ صادق در نامساوی

$$|s_i^{j_i}| \geq u \max\{|s_i^k|, 1 \leq k \leq n - i + 1\},$$

متناظر با سطری از $H_i$ باشد که دارای کمترین تعداد عناصر غیر صفر است(در صورتی که کمینه روی چند اندیس رخ دهد اولین را انتخاب کنید).

**مرحله ۵.** $s_i$ و $G_i$ را به صورت گفته شده در نظر بگیرید و قرار دهید $H_{i+1} = G_i H_i$.

**مرحله ۶.** اگر $i = m$، متوقف شوید ($H_{m+1}$ یک پایه ی پوچ برای فضای پوچ ماتریس $A$ است). در غیر این صورت قرار دهید $r_{i+1} = r_i - 1$، $i = i + 1$ و به گام دوم بروید.