

این مساله را می‌توان به مسئله کوتاه‌ترین مسیر بین هر دو نقطه گراف تشبیه کرد که آن را با استفاده از الگوریتم فلوید حل می‌کنیم. می‌توانیم با تغییر شرط آن به الگوریتمی دست‌یابیم که این مسئله را برای ما حل کند. فرض می‌کنیم در روند حل مسئله و در مرحله k ام الگوریتم فلوید هستیم، اگر از راس i به راس j بدون گذر از راس k مسیری باشد، پهنای باند آن را $b_{ij}^{(k-1)}$ می‌نامیم. حال اگر با استفاده از راس k به عنوان راس میانی، مسیری از راس i به j پیدا کنیم، پهنای باند آن برابر است با مسیری که پهنای باند آن کمتر است یا $\min\{b_{ik}^{(k-1)}, b_{kj}^{(k-1)}\}$ بنابراین حداکثر پهنای باند میان این دو راس در مرحله k ام بصورت زیر است:

$$b_{ij}^{(k)} = \max\{b_{ij}^{(k-1)}, \min\{b_{ik}^{(k-1)}, b_{kj}^{(k-1)}\}\}$$

حال با توجه به این شرط و الگوریتم فلوید، پیاده‌سازی این الگوریتم به شکل زیر خواهد بود:

```
def max_bandwith(table):
    for k in range(n):
        for i in range(n):
            for j in range(n):
                table[i][j] = max(D[i][j], min(table[i][k], table[k][j]))
```

برای بدست آوردن مقادیر صحیح، پهنای باند هر راس و خودش را بی‌نهایت و پهنای باند دو راس که مسیری بینشان نیست را صفر در نظر می‌گیریم. از طرفی می‌توانیم با استفاده از الگوریتم دایکسترا، این مقادیر را با تغییر شرط الگوریتم و برای هر گره جداگانه محاسبه و در ماتریس ذخیره کنیم.