

الگوریتم تقریبی پیشنهادی به این صورت عمل می‌کند که ابتدا  $n$  قطعه را به صورت نزولی مرتب می‌کند و سپس از سنگین‌ترین قطعه شروع می‌کند و به نوبت قطعات را به کامیون‌ها اختصاص می‌دهد. به این صورت که هر قطعه را به کامیونی اختصاص می‌دهد که تا الان وزن کمتری به آن اختصاص داده شده است. یک اجرای الگوریتم در زیر آمده است که در آن جواب الگوریتم ۱۴ شده است که برابر با جواب بهینه نیز هست:

```
C: > Users > Asus > Desktop > projects > post.py
1  w = [5,4,8,2,6,9,1,2,4,6,8]
2  k = 4

TERMINAL  PROBLEMS  OUTPUT  DEBUG CONSOLE

PS C:\Users\Asus> & C:/Users/Asus/anaconda
[9, 8, 8, 6, 6, 5, 4, 4, 2, 2, 1]
[14, 13, 14, 14]
14
```

زمان اجرای الگوریتم نیز  $O(nk)$  می‌باشد؛ زیرا حلقه  $\text{for } n$  بار تکرار می‌شود و در هر تکرار سبک‌ترین کامیون انتخاب می‌شود که  $O(k)$  است.

اثبات ضریب تقریب: فرض کنیم وزن قطعات به صورت مرتب شده  $w_1 \geq w_2 \geq \dots \geq w_n$  است. در این صورت برای مقدار بهینه  $Opt$  روابط زیر را خواهیم داشت:

$$Opt \geq w_1$$

$$Opt \geq \frac{1}{k} \sum_{i=1}^n w_i$$

از طرفی چون وزن‌ها نزولی مرتب شده‌اند داریم:

$$w_j \leq \frac{1}{j} \sum_{i=1}^j w_i$$

حال اگر  $n \leq k$  باشد که بدیهتا الگوریتم پیشنهادی جواب بهینه را می‌دهد. پس فرض می‌کنیم  $n > k$ .

حال بدون کاستن از کلیت فرض می‌کنیم در الگوریتم پیشنهادی، وزن اختصاص داده شده به کامیون اول از همه بیشتر است و  $j$  اندیس آخرین قطعه‌ای است که به این کامیون داده شده است. اگر  $j \leq k$  باشد به این معنی

است که به این کامیون تنها یک قطعه داده شده است. زیرا در ابتدای الگوریتم که کامیون‌ها خالی هستند به هر کدام یک قطعه داده می‌شود. در این صورت هم الگوریتم جواب بهینه را می‌دهد. پس فرض می‌کنیم  $j > k$ . اگر جواب الگوریتم (یعنی وزن کامیون اول بعد از اختصاص دادن همه قطعات) را  $A$  فرض کنیم خواهیم داشت:

$$\frac{1}{k} \sum_{i=1}^{j-1} w_i \geq (A - w_j)$$

زیرا هنگام اختصاص دادن قطعه‌ی  $j$ ام کامیون اول از همه سبک‌تر بوده است. پس خواهیم داشت:

$$Opt \geq A - w_j$$

و در نتیجه:

$$A - Opt \leq w_j \leq w_1 \leq Opt \Rightarrow A \leq 2 \times Opt$$

که به این معنی است که ضریب تقریب الگوریتم ۲ است.