Schemat aproksymacyjny to rodzina algorytmów $\{A_\varepsilon\}$ taka, że dla każdego $\varepsilon > 0$ algorytm A_ε dostarcza rozwiązania problemu z błędem nie większym niż ε, czyli dla problemu ekstremalizacji funkcji $K(x)$, $x \in X$, mamy:

$$|K^{A_\varepsilon} - K^*| \leq \varepsilon K^*,$$

gdzie K^* jest wartością optymalną funkcji K, a K^{A_ε} jest wartością funkcji K dla rozwiązania dostarczonego przez algorytm A_ε.

Wyróżniamy dwa typy schematów aproksymacyjnych:

1. Wielomianowe schematy aproksymacyjne (PTAS): czas działania algorytmu A_ε jest ograniczony od góry przez wielomian zależny od rozmiaru instancji $N(I)$ oraz funkcję wykładniczą zależną od odwrotności błędu $1/\varepsilon$.
2. W pełni wielomianowe schematy aproksymacyjne (FPTAS): czas działania algorytmu $A\varepsilon$ jest ograniczony od góry przez wielomian zależny od rozmiaru instancji $N(I)$ oraz wielomian zależny od odwrotności błędu $1/\varepsilon$.

PTAS dla $P2||C_{max}$:

Procedure PTAS(n,k)

1. Posortuj zadania wg nierosnącego porządku czasów ich wykonywania p_j ($p_1 \geq p_2 \geq \ldots \geq p_n$), tworząc listę zadań L – złożoność $O(n \log n)$
2. Przyporządkuj k pierwszych zadań ($1 \leq k \leq n$) z listy L do procesorów w sposób optymalny (przegląd zupełny), ignorując zadania pozostałe – złożoność $O(2\char`^k)$
3. Pozostałe ($n - k$) zadań przyporządkuj kolejno do najwcześniej dostępnego procesora – złożoność $O(n)$

Ostatecznie, złożoność obliczeniowa alg. PTAS wynosi $O(n \log n + 2^k)$.
Błąd jest odwrotnie proporcjonalny do k, czyli $\varepsilon \sim 1/k$.
W efekcie: wraz ze wzrostem k rośnie złożoność (czas działania) algorytmu (wykładniczo!), ale maleje błąd (proporcjonalnie). Można stwierdzić, że $k=O(1/\varepsilon)$, co daje złożoność obliczeniową alg. PTAS zgodną z definicją, tzn. $O(n \log n + 2\char`^\{1/\varepsilon\})$

Do skonstruowania FPTAS dla problemu $P2||C_{max}$ wykorzystamy algorytm programowania dynamicznego o złożoności (pseudowielomianowej) $O(n \cdot S)$, gdzie $S = \sum_{j=1}^{n} p_j$

Procedure FPTAS(n,k)

1. Utwórz nową instancję problemu $P2||C_{max}$, podstawiając $\widetilde{p_j} = \left\lfloor \frac{p_j}{k} \right\rfloor$ dla j=1,...,n oraz k jest liczbą całkowitą większą od 1. -Złożoność $O(n)$
2. Rozwiąż nową instancję problemu metodą programowania dynamicznego (optymalnie). – Złożoność $O(nS/k)$
3. Rozwiązanie uzyskane w Kroku 1 zastosuj do pierwotnej instancji problemu (będzie ono rozwiązaniem przybliżonym!). – Złożoność $O(n)$

Ostatecznie, złożoność obliczeniowa alg. FPTAS wynosi $O(nS/k)$ (mniejsza niż $O(nS)$.
Błąd jest proporcjonalny do k, czyli $\varepsilon \sim k$.
W efekcie: wraz ze wzrostem k maleje złożoność (czas działania) algorytmu (wykładniczo!), ale rośnie błąd (proporcjonalnie). Można stwierdzić, że $k = \left\lfloor \frac{\varepsilon S}{2n} \right\rfloor$ co daje złożoność obliczeniową alg. FPTAS zgodną z definicją, tzn. $O(n^2/\varepsilon)$.

Podsumowując:
Schematy aproksymacyjne są algorytmami przybliżonymi (tzn. dostarczają rozwiązań nieoptymalnych). Ich podstawową cechą jest możliwość wpływania na maksymalny błąd uzyskiwanych wyników.
Niestety odbywa się to kosztem czasu działania (im mniejszy dopuszczalny błąd tym dłuższy czas działania).
PTAS - silna zależność błędów od czasów (stąd praktyczne ich zastosowanie jest ograniczone jedynie do przypadków, w których dopuszczamy stosunkowo duży błąd rozwiązania).
FPTAS - większa skuteczność niż PTAS (możliwość uzyskania rezultatów o mniejszym błędzie w rozsądnym czasie).
Zostało udowodnione, że nie dla wszystkich problemów NP-trudnych można skonstruować¢ schemat aproksymacyjny (o ile $P \neq NP$). Ponadto, FPTAS można skonstruować¢ tylko dla problemów NP-trudnych w zwykłym sensie (nie silnie NP-trudnych).
Błąd względny algorytmu LSA jest nie większy niż $1-1/m$, co dla dwóch maszyn daje 50%, dla m = 3 mamy 66% (to maksymalne wartości, zazwyczaj błąd jest mniejszy)
Błąd względny algorytmu LPT jest nie większy niż $4/3-1/3m$, co dla dwóch maszyn daje 16%, dla m = 3 mamy 22% (to maksymalne wartości, zazwyczaj błąd jest mniejszy)