

Az algoritmus egy távolság tömböt használ, amelyben elmenti a működés közben talált legrövidebb utakat minden csúcshoz. Minden lépésben veszi a még nem "kész" csúcsok közül a legközelebbit és "kész"-nek jelöli. Ezek után megvizsgálja, hogy ezen csúcson keresztül nincs-e javító út más csúcsokhoz, és ha talál ilyet azt jelzi a tömbben.

A probléma párhuzamosítását azzal kell, hogy kezdjük, hogy a problémateret valamilyen módon alterekre bontsuk ahhoz, hogy párhuzamosan végezhessünk rajtuk számításokat. Sajnos ezt triviális módon nem tudjuk megtenni, mert nincsenek egyértelmű független részei az algoritmusnak.

A feladatot így a csúcsok szerint fogjuk elosztani. Ez azt jelenti, hogy mindazon helyeken a programban, ahol egy ciklus az összes csúcst sorra veszi átírjuk úgy, hogy az adott ciklus csak a csúcsok egy részhalmazán menjen végig. Ehhez használhatunk ciklikus felbontást (loop scheduling) vagy tömbök szerinti felbontást (block scheduling) akár. Az előbbit választottuk, így tehát programunkban minden ciklust a következőképpen fogunk átírni:

for(i = id; i < N; i += nproc)

A globális minimumot párhuzamosan a következőképpen tudjuk meghatározni. Először minden folyamat a lokális minimumot fogja meghatározni azokon a csúcsokon, melyek hozzá vannak rendelve. Ezek után a mester folyamat összegyűjti a lokális minimumokat és kiválasztja közülük a legkisebbet - ez lesz a globális minimum -, majd ezt az információt visszaküldi a "szolgáknak".

Azaz a lokális minimum $D[x]$ és helye x kiszámolása után a kettőből egy párt készítünk. A szolgák elküldik a saját párjukat a mesternek. A mester fogadja a párokat, és minden fogadásnál összeveti az eddig talált és elmentett optimálissal, hogy nem kisebb-e annál a fogadott $D[x]$ érték. Ha igen, akkor a mester kicseréli az eddig elmentett optimumot a frissen fogadottra. A fogadás végén a mester visszaküldi az optimális értékeket és a szolgák frissítik az x és $D[x]$ értékeiket a globális minimummal.

Két mesterséges gráfot készítettem 30 000 illetve 300 000 csúccsal, és ezekre a bemenetekre futtattam a programot különböző processzormagszámra.

| nproc | 30.000 | 300.000 |
|-------|------------|------------|
| 1 | 80.0s | 9841s |
| 12 | 10.8s 7.4x | 2302s 4.3x |
| 24 | 5.7s 14.0x | 1320s 7.5x |
| 48 | 3.2s 25.0x | 447s 22.0x |
| 96 | 2.4s 33.0x | 226s 43.5x |

A megoldás kommunikációs szükségessége a minimum távolságok szinkronizációjára korlátozódik. A mester begyűjti a lokális minimumokat, kiszámolja a globális minimumot és az eredményt broadcast-al visszaküldi a szolgáknak.

A gyorsulást (speed up) a lokális munka és a kommunikáció aránya korlátozza csak, azaz a kérdés az, hogy az elvégzendő munkához képest milyen gyakran van szükség üzenetküldésre.