

それを簡単にご説明しますと、まず最初になんでもいいから順序をきめておいて、そしてその中の相隣りあう2つを頭のほうからひっくり返していくわけです。そして相隣りあう2つをひっくり返したときのいいほうを取りまして、つまり部品の流れがこういうふうにあると、これをひっくり返すとあとどうなるかということ計算する。そしてひっくり返したほうがよければこれをやる。そして次にこれをひっくり返す。そうすると $n \times (n-1)$ ぐらいの程度になりますから、たとえばダイナミック・プログラミングで $n=100$ ならば、10の14乗年ぐらいかかっているのが、そういうふうにすれば1秒ぐらいで済むということで、実現可能ということになります。

そういうふうにして、それぞれをこの J という値でやってみると、たとえば工程が5つで部品が7つぐらいのものだと、もし図面が入ってきた順に作業をやるというやり方、あるいは少し現場でカンをきかせて作業をやるというものに対して、完全列挙法をやった場合は、工程が5で部品が7ぐらいだと完全列挙もできますが、これは式をとってみないとわかりませんが、0.647に対して0.736ですから、20%ぐらいの利得が完全列挙法でやるとあるわけです。この完全列挙法というのは非常に時間がかかるというので、先ほどお話ししたペア・コンパリジョン・メソッドというのをを使うと——完全列挙法だと0.736になるのが、ペア・コンパリジョンだと0.7ぐらいですから、ほとんどいいですね。だから20%ぐらいの利得が17.8%ぐらいの利得になるという程度で、かなりの効果をあげることができるというような結果が得られております。

これは1つだけの例についてお話しましたがけれども、5つ、6つのシミュレーションをしますと、ほとんどその完全列挙とか、あるいはジョンソン・メソッド——工程が2つの場合はジョンソン・メソッドが使えるわけですがけれども、そういうような場合とほとんど一致しているという結果が得られております。だからペアコンパリジョンというのは、かなり簡単でしかも有効な方法である。

そういう方法を使って、いまのように目的函数を設定して、実際に効果をあげたというようなことがありますので、ご参考までにお話したわけです。

どうも非常に雑ばくな話で、厳密な理論は一切お話ししないで、どうやって実際に会社でやって効果があがったかということを中心にしてお話したわけですが、一応この辺で終らせていただきたいと思ひます。

以 上