# ラグランジュ分解・調整法と動的スケジューリング

黒田　充

## 1.　はじめに

近年，**ラグランジュ分解・調整法**（Lagrangian Decomposition and Coordition Method）がその高速性，高品質の解，動的問題に対する適合性の故にスケジューリングの方法として注目を集めている[1]，[2]，[3]．これはラグランジュ緩和法と通常呼ばれているが，組み合わせ問題を分岐限定法で解く際に目的関数の界値を求めて探索空間を縮小するというよく知られた方法と区別するために，標記の名称（以下，LDC法と呼ぶ）を用いることにした．

LDC法が実用的であるとみなされている理由は前述の特性を備えていることにあるが，なかでも動的問題との適合性はひと際目立っており，最適解法の適用を従来静的問題に限定してきたスケジューリング研究の慣行に対するアンチテーゼとしての意味合いがそれにはある．しかも，実際には動的問題の方が普通の問題であって，静的問題の多くは現実問題を解くために便宜上仮定されたものであり，その特性が持っている意味はことさら大きい．本稿では，動的問題の解法，つまり**動的スケジューリング**の方法としてのLDC法の側面に焦点を当てて論じることにしたい．

## 2.　静的問題と動的問題の定義

スケジューリング問題を静的問題と動的問題に分類する試みが従来から行われている．例えば，French[4]はこれらの問題を次のように定義している．

静的問題：解く前に全ての数値が判っていて，その後変更されることのない問題

動的問題：時間の経過につれてジョブがランダムに到着する問題

さらに，確率的問題という分類を別に設け，次のような定義を示している．

確率的問題：作業時間等が不確定な問題

くろだ　みつる　青山学院大学理工学部
〒157-8572 東京都世田谷区千歳台6-16-1

Frenchによるこの分類は，現実のスケジューリング問題を区別するものではなく，スケジューリングのモデルを分類するものであることはそれらの定義から明らかである．つまり，静的問題はすべてのデータが与えられていて変更されることのない問題であり，評価関数が一つであれば最適化の対象になる．一方，動的問題の場合は，計算速度やコンピュータの処理能力を考ると最適解法は現実的でなく，解法としてはもっぱらシミュレーションが利用されてきた．確率的問題は，データの期待値を用いれば静的問題としての取り扱いができ，最適化の対象になり得る．

一方，筆者は現実のスケジューリング問題を分類する観点に立って，静的問題と動的問題の定義を以下のように与えている[5]．

静的問題：スケジュール作成時にスケジューリングに関する全てのデータが既知であって，その後変更されることのない問題

動的問題：スケジューリングに関するデータの一部がスケジュール作成時に未知であったり，スケジュール作成後に変更されることのある問題

この定義によれば，動的問題はデータの追加や変更を前提としたものであり，Frenchが定義している確率的問題はこのクラスに含まれる．静的問題の例としては，1シフト中にすべてのジョブの処理が完了できるような場合に，操業に先立ってスケジューリングを行い，求められたスケジュールにしたがって実際の作業を実施するというものがあげられる．その場合，データが予定されていたものと異なった値をとったとしても，作業に差し支えがない限りその変化は無視されよう．言うまでもなく，作業に差し支えが生じるような変化が生じた場合は，作業現場の判断でスケジュールの修正が行われる．

## 3. 動的スケジューリング問題の例

ごく一般的な動的問題はジョブの処理時間がシフト時間より長くなるようなジョブショップにおいて見られる．通常，1シフト以上時間が経過するとデータの変更や追加が生じるし，データの予定値と実際値の差が累積され，最初に立てたスケジュールが役に立たなくなってしまう．このような場合，1シフト毎にジョブとショップに関する最新のデータを用いてスケジュールの更新が行われる．つまり，現シフトのスケジュールのみが作業にあたって利用され，それ以後のシフトのスケジュールは作成されるけれどもそれにしたがって作業が行われることはなく，将来のジョブの進捗状況やショップの稼動状況を予想する資料として利用される．この種のスケジューリングを表す名称として，**ローリング・スケジューリング**がある．

ローリング・スケジューリングにおいてスケジューリングの更新間隔を短くするとどうなるであろうか．スケジュールが現状を正しく反映していて，しかもうまく作られているならば，ショップのパフォーマンスは当然改善される．しかし，実際にこれが行われないのは，現状をスケジューリングに反映することが困難であったり，問題の規模が大きくて計算時間がかかり過ぎたりするためである．例外として，FMSのようにワークの搬送と加工が自動的に行われ，問題の規模もそれほど大きくない場合，ジョブとショップの状況は正確に把握できるので更新間隔を短くすることに意味があろう．とは言え，スケジューリングに時間をかなり要する場合，更新間隔には限度がある．そこでスケジューリングを最後まで行うのではなく，変化が生じる都度，当面必要とする部分に限定してスケジュールが求められる．多くは，前もって定めておいたディスパッチング・ルールを用いて，加工の終了した機械において次に加工するワークを決めたり，加工の終了したワークを搬送する装置を選んだりする．このようなスケジューリングのやり方は**リアルタイム・スケジューリング**と呼ばれる．

## 4. ラグランジュ分解・調整法の現実的意味

LDC法によるスケジューリングは，生産管理の伝統的な手法である山積み・山崩しに類似している．それらの相違点は，数学的に行うか人間の判断を用いて行うかの違いであると言えるくらいである．ただ，山積・山崩法は負荷計画に用いる方法であるため，負荷の対象期間として1日や1シフトのような長い時間が取り上げられ，そのような単位期間中に複数のジョブやジョブを構成する作業が割り付けられるのに対し，LDC法の場合は，スケジューリングであるので，はるかに短い時間，例えば1分とか30秒のような短い時間を単位として表される時間帯にジョブを構成している作業が割り付けられ，負荷計画のように作業の実行順序を決めないでそのままにしておくことはない．この単位時間はLDC法で重要な役割を担っており，本稿ではこれを**タイムスロット**と呼ぶことにする．

LDC法では，基本的に，あるジョブを構成する作業を他のジョブの割り付け状況を無視してそれぞれのジョブにとって最も望ましいタイムスロットを選んで割り付ける（図1参照）．これは機械や設備の能力を無視した割り付けを行うことに他ならず，結果としてオーバーロードが生じる．オーバーロードの発生を許容し，後で平準化してその解消を行う点が，山積・山崩法と瓜二つなのである．

このような制約条件の違反を計算の過程で認める考え方を，数学的に取り扱うものがラグランジュ緩和法であることはよく知られている．より具体的に述べるために，納期遅れの総和を最小化するジョブショップ・スケジューリング問題を取り上げよう．いま，ジョブを一つずつ取り上げて作業間の先行関係が満たされるようにタイムスロットへの割り付けを繰り返し，そのままでは実行不能なある生産スケジュールを得たとしよう．このスケジュールの評価関数をラグランジュ関数によって表すと次のようになる．

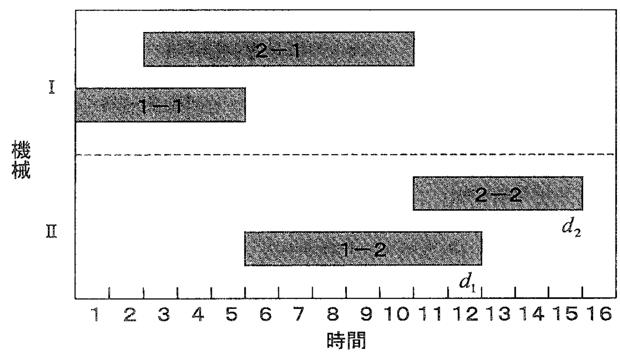


**図1** オーバーロードの生じた実行不能スケジュール

例えば，ショップに機械Iが2台，機械IIが1台あるとしよう．それぞれ二つの作業から成るジョブ1，2をこのガンとチャートが示すように割り付けをすると，機械IIでタイムスロット11と12においてオーバーロードが生じる．

$$L(c,\lambda)=\sum_j f_j(c_j)+\sum_k\sum_m \lambda_{km}\left(\sum_j \delta_{jkm}-M_{km}\right) \qquad (1)$$

ここで，

$c_j$：ジョブ$j$の最後の作業が終了するタイムスロット

$f_j(c_j)$：ジョブ$j$が納期に遅れた場合に発生する罰金の額

$\lambda_{km}$：タイムスロット$k$における機械$m$のオーバーロードに対して支払われるべき機械1台あたりの価格（使用料）

$\delta_{jkm}$：ジョブ$j$の作業のいずれかがタイムスロット$k$における機械$m$に割り付けられている場合は1，割り付けられていない場合0をとる0-1変数

$M_{km}$：タイムスロット$k$において稼動可能な機械$m$の台数（$m$は機械の種類）

したがって，ラグランジュ関数$L(c,\lambda)$はスケジューリングの対象になっているすべてのジョブに関する納期遅れの罰金額と不足した機械の使用料の総和を表している．$\lambda_{km}(\geq 0)$はラグランジュ乗数あるいはラグランジュ変数と呼ばれるもので，初期値を除けば，その値は計算の結果として求められる．

スケジューリングの過程では，所与の$\lambda_{km}$の下で$c_j$を変数として$L(c,\lambda)$を最小化する計算と所与の$c_j$の下で$\lambda_{km}$を変数として$L(c,\lambda)$を最大化する計算が交互に繰り返される．ただし，最小化や最大化はなるべく小さくする，あるいは大きくするという程度の意味しかなく，繰り返し過程中の各段階における厳密な最適化は不要である．しかしながら，繰り返し過程の終了時には，本来の目的関数，この場合であれば総納期遅れを近似的に最小化するスケジュールがその精度を保証して求められる．

LDC法を用いる場合，ラグランジュ変数$\lambda$の値と最適化に要する時間の間には密接な関係があることが知られている．$\lambda$の値は繰り返し計算が行われる都度，最終的な値$\lambda^*$に少しずつ接近していくため，$\lambda^*$に近い$\lambda$から計算を開始すると，繰り返し回数が少なくなり計算時間は大幅に削減できる．この事実はLDC法の動的問題への適合性を意味するものであり，データの変化が生じたときに最新の$\lambda^*$を初期値として再最適化計算を行うことによって高い計算効率が保証される．

その上，$\lambda^*$と求められた生産スケジュールは1対1の対応関係があるため，最新の$\lambda^*$を用いて再最適化計算をすると，ショップ全体に影響を及ぼす変化が生じない限り，これから求める生産スケジュールが，変化が生じる直前に求めたものと比べて全く異なるようなことはあり得ない．つまり，変動するショップの環境に適応するとともに，作業実施の観点から見れば継続性のあるスケジュールが作られるという意味の解の高品質性が保証される．

## 5．リアルタイム・スケジューリングの擬似最適化

リアルタイム・スケジューリングの方法としてよく用いられるのはディスパッチング・ルールであることはすでに述べた．確かに計算労力の無駄は生じないし，どのような問題にも適用できるという点でこれ以上に頑健なスケジューリングの方法は存在しない．しかし，LDC法もまた，この動的問題の極限に位置付けられるリアルタイム・スケジューリングに適用することが可能である．しかも，ただ使えるというだけでなく，徹底的な利用にも堪える可能性を秘めている．

そこで，**リアルタイム・スケジューリングの擬似最適化**（quasi-optimization）と称する試行を企てる．これは，動的問題をデータが一時的に固定された擬似静的問題の連鎖とみなし，新しい擬似静的問題が定義される都度，精度の保証された最適解を求め，その解つまりスケジュールに従って作業を実施するというものである．この場合，予定されていない変化が生じると，擬似静的問題が定義し直され，直ちに再最適化が行われる．ただし，変化は必ずタイムスロットとそれに続くタイムスロットの間で生じるものとする．図2にLDC法を用いたリアルタイム・スケジューリングの方法が示されている[5][6]．

図中のランダム事象は予定されていなかった事象を指し，ジョブの飛び込み（特急作業），納期の変更，設計変更による加工経路や作業時間の変化，作業の遅延，機械故障の発生などはその代表的なものである．その他に，周期的に発行される製造命令書もその内容が事前に知られていない限り，疑似静的問題の再定義を必要とするためランダム事象として取り扱う．つまり，計画の実現可能性や最適性を損なう可能性のある変化はすべてランダム事象であると見なされる．

リアルタイム・スケジューリングの環境下では，ランダム事象がどの程度頻繁に発生するかによってリスケジューリングに要する計算負荷が異なる．そこで，

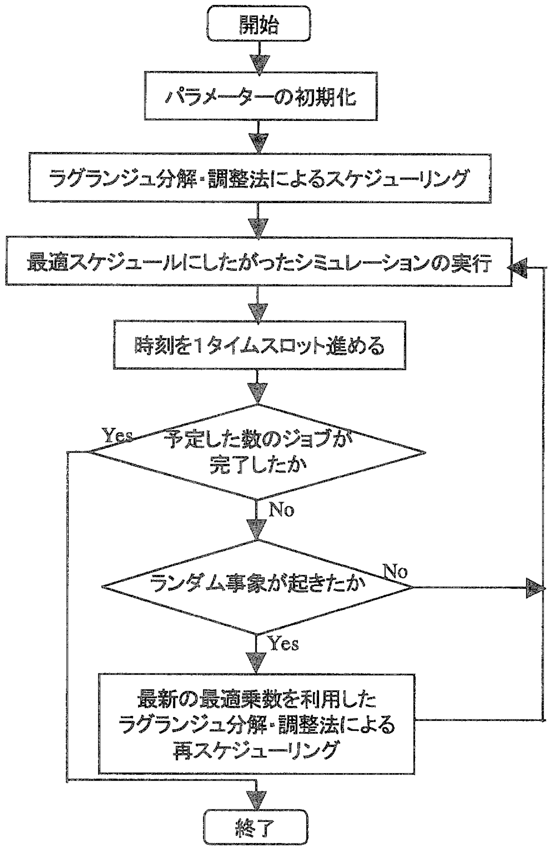


図2　リアルタイム・スケジューリングの疑似最適化

動的問題の特性を示す普遍的な尺度として，ランダム事象の発生密度を定義する．

いま，製造命令書の発行周期，例えば1シフトを $t$ で表す．問題は1周期 $t$ の間にどのくらいの頻度でランダム事象が発生するかであり，それについては見積もりが必要である．これは期待値であるから，整数でなくてもよく，例えば2.5回というような値を定める．また，これはパラメータであるから，一つの値である必要はない．この値を $f$ で表すと，ランダム事象の発生密度 $d$ は次式で与えられる．

$$d=(f+1)/t \tag{2}$$

この分子の値がそのまま1周期中にリスケジューリングを行う回数の期待値になる．

以下，筆者らが行った数値実験について紹介しよう．表1はモデルの基本的な条件を要約したものであり，ワークセンタが3箇所，それぞれに配置された機械が2台，計6台の機械から成る小さなジョブショップを想定している．ジョブは10タイムスロット毎に1件が到着し，いずれも3つの作業を必要としており，各ワークセンタに均等に負荷がかかるように条件が設定されている．加工経路，作業時間，納期とも一定の規則にしたがってランダムに決定される．

表1　ジョブショップ・スケジューリング・モデルの基本的条件

| | |
|---|---|
| ①初期のジョブ数 | 10 |
| ②到着間隔($\Delta t$) | 10 |
| ③到着ジョブ数/1回当たり | 1 |
| ④計算終了基準を与えるショブ完了数 | 150 |
| ⑤ワークセンタ数 | 3 |
| ⑥機械台数/ワークセンタ当たり | 2 |
| ⑦作業数/ジョブ当たり | 3 |
| ⑧加工経路　以下の6経路からランダムに抽出 | (1-2-3) (1-3-2) (2-1-3) (2-3-1) (3-1-2) (3-2-1) |
| ⑨処理時間($\Delta t$) | U(1～35) |
| ⑩納期($\Delta t$) | 到着時刻＋$\alpha\Sigma$U(1～35) |
| ⑪納期係数 $\alpha$ | U(1～3) |

* $\Delta t$：タイムスロット
** U(a～b)：範囲[a, b]，平均値(a+b)/2の一様分布
*** 負荷率：90%；(18*3/10*3*2＝0.9)

表2　納期変更の条件

| | |
|---|---|
| ①納期変更の発生頻度(回/到着間隔) | 以下の5通りについて実行 0.0，0.25，0.5，1.0，2.0 |
| ②納期の変更幅($\Delta t$) | 以下の4つの変更幅の中からランダムに抽出 −8，−4，+4，+8 |
| ③納期変更が行われるジョブの割合(%) | 以下の3通りについて実行 10，20，30 |

本モデルで取り上げたもう一つのランダム事象である納期変更に関する条件が表2にまとめられている．①はパラメータの一つである納期変更の発生頻度を示しており，ランダム事象の発生密度を(2)式によって求めると，0.100，0.125，0.150，0.200，0.300になる．②に示された納期の変更幅はランダムに選出され，マイナスの場合は特急作業への変更を意味している．③は納期変更に関するもう一つのパラメータであり，ショップ中のジョブのそれぞれがランごとに一定の割合で選出され，納期変更が行われる．ただし，一つのジョブにつき2度以上納期が変更されることはない．この納期変更の割合を“ランダム事象の強さ”と呼び，“ランダム事象の密度”とともに計算時間に影響する要因として数値実験結果の考察で取り上げる．

以下に計算結果を示そう．図3はディスパッチング・ルール（Slack）を用いたシミュレーションとLDC法の間で目的関数がどの程度異なるかを表している．目的関数は総納期遅れであるから，評価値は小さい方が良い．興味深いのはランダム事象の発生密度，ランダム事象の強さともに評価値にあまり影響しない

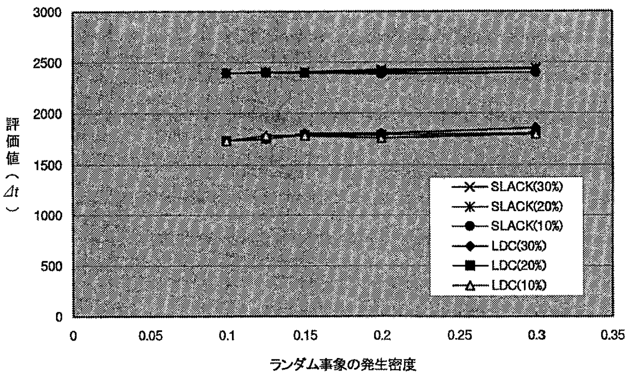


図3　シミュレーションとLDC法間の目的関数値の比較

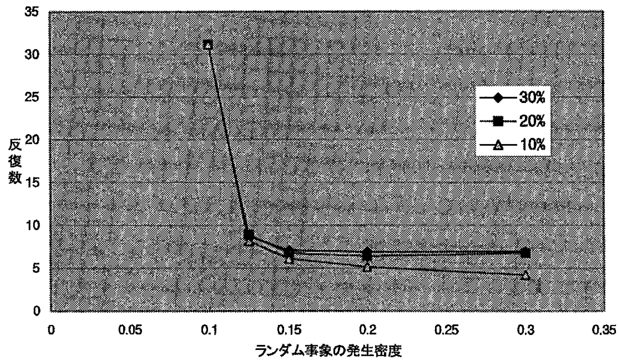


図4　発生密度と最適化1回当たりの反復数の関係

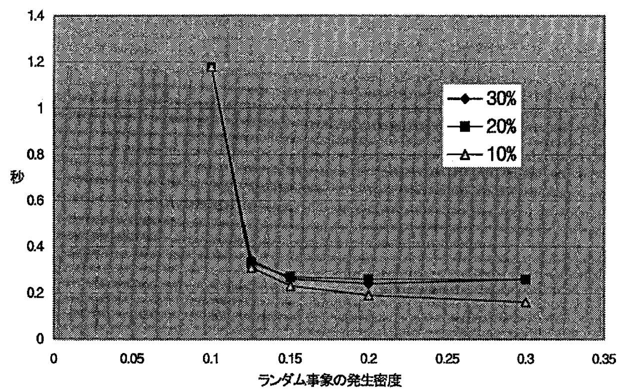


図5　発生密度と最適化1回当たりの計算時間の関係

という点である．

図4と図5には，ランダム事象の増加とともに再最適化に要する反復数と1回当たりの計算時間（平均値）が減少する様子が描かれている．これらよりいずれも一定の値に収束するとともに，ランダム事象の強さが大きいほど高い値に収束する傾向が見られる．なお，使用機種はIBMPC 750（Pentium　99 MHz，32 Mメモリ搭載）である．

ところで，1タイムスロット中に再最適化のためにどのくらいの計算負荷がかかるのであろうか．(2)式で

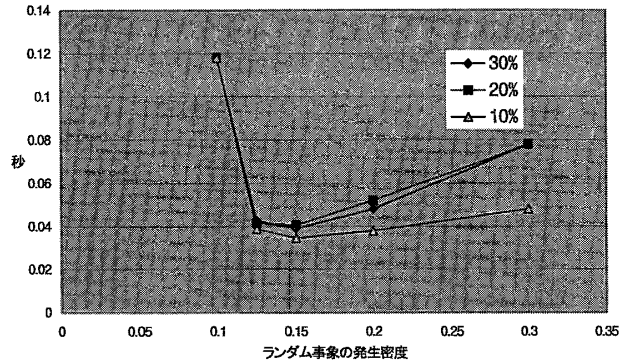


図6　発生密度とタイムスロット当たりの計算負荷の関係

例えば，ランダム事象の発生密度が0.3，タイムスロットが60秒の場合，再最適化のための計算負荷は平均して0.1秒より小さいから，リアルタイム・スケジューリングの擬似最適化は可能である．

示されるランダム事象の発生密度 $d$ は，製造命令の発行周期 $t$ 中に発生するランダム事象の発生回数を $t$ で割った値，つまりタイムスロット中に実施される再最適化の回数の期待値を意味するため，1タイムスロット $\Delta t$ 当たりの計算負荷（$CWL/\Delta t$）は次式で与えられる．

$$CWL/\Delta t=\bar{m}\times d \tag{3}$$

ここで，

$\bar{m}$：再最適化1回当たり平均計算時間の収束値

$d$：ランダム事象の発生密度，つまりタイムスロット当たりに必要な再最適化の回数の期待値

したがって，$\bar{m}$ の見積もりが可能であれば，計算負荷のおおよその予測ができ，それがタイムスロットより小さければ，LDC法によるリアルタイム・スケジューリングの擬似最適化は理論上可能である（図6参照）．スケジューリング問題の規模は（対象タイムスロット数×機械の種類数）で表せて大きな数になるが，通常の組み合わせ問題の解法と異なって，LDC法の場合，計算時間の指数的増加は起こり得ないので，計算機を特定するならば，いく通りかの規模の問題を取り上げて数値実験を行えば，かなり高い精度で $\bar{m}$ の値の見積もりができよう．

## 6.　座席予約型生産スケジューリングへの応用

動的問題の中で現在最も注目されているものがAPS（Advanced Planning Scheduling）の主要部分を構成しているスケジューリングであろう．APSは従来から行われている部品展開と資材の手配をスケジ

ューリングと同期させて行う技術であり，短納期を実現することにその狙いがある．したがって，APSの先端的な利用の仕方は次のようになる．

（1） 顧客からの引き合いがあると，生産現場の負荷状況を考慮した製造シミュレーションを実施し，納期見積もりを行う．その際に，必要ならば主要な資材（中間製品）をその仮のオーダの紐付きにする．

（2） 顧客が提示された納期を受け入れると，製造シミュレーションの結果，予定された生産日程が原則的に顧客に対して保証される．つまり，生産設備が，必要な時期の望ましい期間，そのオーダのために予約される．また，前述した資材がそのオーダのために確保され，同時にその他の必要な資材が確定した生産日程に基づいて手配される．

この製造シミュレーションは，納期に最も影響する（ネック工程を持っている）生産現場を中心として行われるが，これは引き合い中の顧客のオーダを考慮した仮の生産スケジュールを作ることに他ならない．APSにおいて行われるこのスケジューリングは，従来の生産スケジューリングと次の点で異なっている．

顧客からの引き合いはランダムに生じることを前提としており，受注が一たび確定すると納期は保証されるため，受注済みのオーダの生産スケジュールに影響を及ぼさないように新規に到着したオーダをすでに作られているスケジュールに組み込む必要がある．その上，納期（生産リードタイム）ができる限り短くなるようなスケジュールを作ることが要求される．このようなスケジューリングを顧客からの引き合いがある度に繰り返して行う．

この新種の動的問題を従来のスケジューリングと区別するために**座席予約型生産スケジューリング**と呼んでおり，座席は生産設備と主要な資材を指し，短納期を保証する上で欠かせない概念になっている．以下において，LDC法の座席予約型生産スケジューリングへの適用法を述べる．

LDC法が前述の目的に対して最も効力を発揮するのは生産現場がジョブショップの場合であるから，対象としてジョブショップを想定して説明する．

LDC法がいま述べた難しいスケジューリング問題に対して他のスケジューリングの方法，例えばディスパッチング・ルールを用いたシミュレーションに比べて優位性を持っているとは言え，受注済みのオーダに関するスケジュールを完全に固定して再最適化を行うと多くの場合に良い結果は得られない．そこで，スケジューリング・レベルにおいて納期（加工終了の時期）を少しずらすことを考える．どれだけずらせるかは，職場や製品の特徴によって決まるものであるが，実際の納期が変更されるようなことがあってはならない．スケジュールの時間単位がタイムスロットという短い時間であることを述べたが，実際の納期は通常それよりも大きな時間単位で示されるからその時間単位の違いを利用して行えば良い．これは再最適化に際して，受注済みのオーダの評価関数 $f_j$ を定義し直すだけで行える（図7参照）．また，すべての $f_j$ を変更する必要もない．

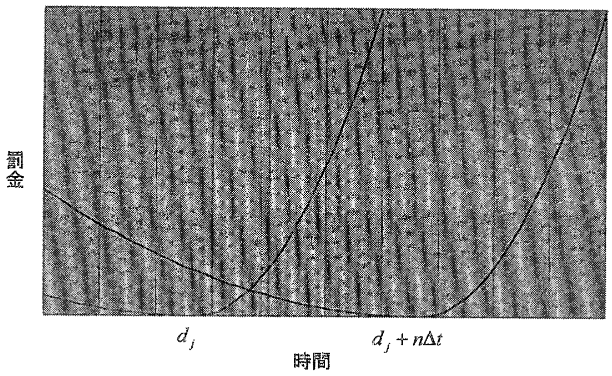


図7 評価関数 $f_j$ の調整による振動法の実施

後は，計算が実行されて機械の空き時間を活用した効率の良いスケジュールが自動的に作成されよう．これは，砂利を砂こしに入れてゆさぶると砂こしの目より小さい砂利が下に落ちるのに似ているため，**振動法**と名付けている．$f_j$ の再定義の仕方は無数にあるので，対話形式で行えるようにプログラムを作っておくと便利であろう．

振動法を用いても顧客にとって満足の行く納期が見積もれない場合は，実行可能解が得られない状況が継続するため計算を打ち切る必要がある．その際に入手した実行不能スケジュールでは，生産能力を超えた負荷の山積みがいずれかの機械のどこかのタイムスロットにおいて生じており，それに対応する $\lambda_{km}$ が必ず正の値をとっている．職場全体で正の $\lambda_{km}$ がどのくらいあるかを検索して，オーバーロードの箇所を確認することにより，残業によって対処できる場合はその費用の見積もりができる．残業費がかさんでも顧客の要求する納期を条件に受注した方がよいかどうかという判断が望まれる場合は，戦略と採算の両面について検討する必要があり，LDC法にはこの採算面の検討に役立つ資料を提供する機能もあることを留意しておきたい．

## 7. おわりに

以上，ラグランジュ分解・調整法が動的スケジューリングに適した特性を持っていることを述べた．本稿では，この方法が備えている高速性の説明を紙面の都合により割愛したが，これについては他の文献を読んで補っていただければ幸いである．冒頭に述べた高速性，高品質の解，動的問題に対する適合性がそろっている点が肝心なのであり，この意味で本方法が実際のスケジューリングに利用できるようになりつつあるという現実は，スケジューリング研究の長い歴史において画期的な事柄であると言わざるをえない．ラグランジュ分解・調整法が持っている優れた特性を実際のスケジューリングに生かすためには多くの課題が残されており，本領域の研究者に寄せられる期待は大きい．

### 参考文献

[1] Luh, P. B. et al: “Schedule generation and reconfiguration for parallel machines”, IEEE Trans. on Robotics and Automation, Vol. 6, No. 6, pp. 687-696 (1990)

[2] Hoitomt, D. J. et al: “A practical approach to job-shop scheduling problems”, IEEE Trans. on Robotics and Automation, Vol. 9, No. 1 pp. 1-13 (1993)

[3] 米田清：ラグランジュ緩和法によるスケジューリング，システム/制御/情報，Vol. 41, No. 4, pp. 130-138 (1997)

[4] French. S.: Sequencing and Scheduling, An Introduction to the Mathematics of the Job-Shop, p. 16 (1982)

[5] Kuroda, M. and Enomoto, M: “Usefulness of Lagrangian relaxation approach to production scheduling under a dynamic environment”, 1998 JAPAN-U. S. A. Symposium on Flexible Automation, Vol. 2, pp. 899-905 (1998)

[6] 黒田充, 遠国祐介, 榎本雅夫：ラグランジュ緩和法を用いたリアルタイム・スケジューリングの疑似最適化, 生産スケジューリング・シンポジウム’98 講演論文集, スケジューリング学会, pp. 65-70 (1998)