

ユーザの位置情報を利用したサービスのタスクスケジューリングモデル

M-48 Location based Task Scheduling for Ubiquitous Applications

鈴木 源太[†]
Genta Suzuki

岩谷 晶子[‡]
Akiko Iwaya

権藤 俊一[‡]
Shunichi Gondo

徳田 英幸^{†‡}
Hideyuki Tokuda

1. はじめに

利用可能な場所が限定されているサービス[§]がある。それは、サービスを提供する機器公共の場所に設置され携帯不可能なサービス、もしくは特定の場所で利用することに価値のあるサービスである。サービス例として、プリンタやドライブスルーがあげられる。このようなサービスの利用可能場所は、プリンタなどの機器が存在する場所に限られる。ネットワークを利用しない場合、ユーザはこのようなサービスを利用するために、サービスを提供する機器が存在する場所へ移動し、機器に直接サービス要求を発行して、サービスを利用する。この場合、ユーザの要求に応じた処理（以後、タスク）を実行することで利用可能となるサービスでは、サービス要求からサービス利用までのタスクの実行時間が待機時間として課される。さらに、複数のユーザが同一のサービスを利用する場合には、他のユーザのサービス要求に対するタスクの実行時間が待機時間に加えられる。

このようなサービスに生じる待機時間を短縮するために、ネットワークを利用して、遠隔からサービスに対してサービス要求を発行する方法が考えられる。サービス要求を受け取った時点から、サービス側がタスクを開始することで、ユーザがサービスを利用可能な場所に到着後の待機時間を短縮できる。

しかし、サービス要求を遠隔から受けとらない場合と同様に、サービスがサービス要求を受け取った順番でタスクを実行するならば、問題が発生する。たとえば、機器から離れた場所にいるユーザのタスク実行のために、ハードウェアの傍らにいるユーザが余計に待たされてしまう場合である。

上記の問題を解決するために、ユーザの位置情報や移動状況を考慮し、ユーザのサービス利用可能場所への到着に合わせたサービスのタスクスケジューリングを行う必要がある。

2. 本研究の目的

前節で述べたサービスは、タスクの実行性質によって分類できる。タスク実行の性質に応じて、適用すべきスケジューリングモデルは変化する。本研究は、特に以下のタスク実行性質を持つサービスを対象とする。

- タスクの実行時間が予め計算できる
- サービス要求一個につき、実行すべきタスク一個
- サービスは同時に複数のタスクを実行できない
- サービスは一個のタスクをそれ以上分割できない

タスクが分割できないとは、あるタスクの実行開始から実行完了までの間で、中断、再開ができないことである。このようなサービスとして、プリンタや食品の自動調理販売機のようなものが想定される。

これらのサービスに対し、本研究ではサービス利用可能場所に到着後のユーザの待機時間を短縮するスケジューリングモデルを提案する。複数のユーザが利用する場合、全ユーザの待機時間の総和を短縮するポリシーでスケジューリングを行う。

3. タスクスケジューリングモデルの設計

本節では、前節の目的を達成するスケジューリングモデルを提案する。サービス利用可能場所へのユーザの到着予測時刻をデッドラインとして設定し、その時刻までにタスクをスケジューリングする。対象とするサービスはデッドライン後も利用できるため、本スケジューリングモデルはソフトリアルタイムで実現する。

スケジューリングを行う際、ユーザ端末でデッドラインを予測し、それをもとにサービス側で Variable Value Function モデルにもとづいてタスクをスケジューリングする。この作業は再スケジューリングのたびに現在サービスに課せられたすべてのタスクについて行われる。これを行うために以下の二点を設定する必要がある。

- デッドラインの予測法
- 再スケジューリングの実行

はじめに、これらの概要を述べた後、提案するスケジューリングモデルである Variable Value Function モデルの概要を述べる。

3.1 デッドラインの予測法

サービス利用可能場所へのユーザの到着時刻予測であるデッドラインは、ユーザの現在位置から、サービス利用可能場所までの残りの移動距離とユーザの移動速度を使って計算する。残りの移動距離の計算に必要な情報は、ユーザの位置情報、サービスの位置情報、ユーザの移動方向、ユーザの現在位置からサービス利用可能場所までの経路情報である。本研究では、GPS (Global Positioning System) や、加速度センサの機能を搭載した携帯端末をユーザが所持することを想定し、サービス側からデッドラインの要求がある度にユーザ側で前述の必要な情報を収集し、デッドラインを計算するものとする。しかし、このようにして求められるデッドラインには問題点が二つある。以下に問題点とその解決方針を示す。

デッドラインが予測不可能な場合

デッドラインが予測不可能な場合がある。たとえば、ユーザが停止しており、移動速度が0の時である。この問題に対しては、ユーザ側で移動履歴を保持し、スケジューリング更新前の移動速度で次の瞬間から移動するものとしてデッドラインを予測する。最初のスケジューリングの場合は、スケジューリング前の平均移動速度を使ってデッドラインを予測して、解決を図る。

ユーザの移動変化に伴うデッドラインの変更

前述のように、デッドラインはユーザの残り移動距離、ユーザの移動速度をもとに計算する。ユーザが一定の移動速度で、予測経路に沿って進む場合、取得する時間に

[†]慶應義塾大学 環境情報学部

[‡]慶應義塾大学大学院 政策・メディア研究科

[§]本稿においてサービスとは、ソフトウェアのみもしくはソフトウェアと機器の集合体としてユーザに利益をもたらすものである。

かかわらず同一のデッドラインが予測される。しかし、現実世界では移動速度がばらつくことがある。また移動方向、経路選択が必ず予測通りのものになるとは限らない。取得する時刻によって、異なるデッドラインが取得されることがある。

この問題は、実際にユーザが到着するまでは、デッドラインが確定しないため、完全に解決することはできない。本研究では、以後のスケジューリングを決定する時刻(以後、スケジューリング時刻)とデッドラインとの時間差(以後、 Δt)が減少するにつれて、予測したデッドラインが実際のユーザ到着時間に収束していく移動モデルを想定して、デッドラインの変更を考慮したスケジューリングモデルを提案し問題解決を図る。

3.2 再スケジューリングの実行

再スケジューリングの実行には2節で述べたタスク分割ができないというタスクの性質を利用する。あるタスク実行中に、他のタスクのスケジューリングを複数回行ったとしても、タスク実行中の割り込みができないため、結局実行中のタスクの完了を待ってスケジューリングしなければならない。したがって、本研究では再スケジューリングは実行中のタスクが完了する度に行うこととする。また、サービス側にタスクが存在しない場合は、新たなサービス要求が通知された際にスケジューリングを行うこととする。

3.3 Variable Value Function モデル概要

本研究で提案するスケジューリングモデルは、ユーザ待機時間の総和を短縮し、かつデッドラインの変更を考慮したものでなければならない。ユーザ待機時間は、デッドライン後に経過した時間であり、全ユーザの待機時間総和を短縮する順序でタスクをスケジューリングする。待機時間の単調増加に比例して単調減少する Value を想定すると、待機時間の総和短縮は Value の総和の増加と同義である。したがって、待機時間の総和を短縮するためには、このような Value を使う Value Function モデル [1] を適用し、Value を最大とするような順序でタスクをスケジューリングする必要がある。

さらに、本稿ではデッドラインの変更可能性を考慮して、Value Function の傾きと定義域を変化させる Variable Value Function モデルを提案する。Variable Value

である。二つめは、 Δt の閾値の Δt_{th} である。これを超える Δt を持つタスクの Value Function を傾き 0 の直線とする。このとき、タスクはスケジューリングされない。

Δt の変化に応じた Value Function の変化を、図 1 で示す。図 1 においては、Value Function の基本形は (b) の灰線である。あるスケジューリング時刻において、 $\Delta t_1 > \Delta t_{th}$ となるようなデッドライン D_1 が予測された場合、デッドラインの確率分布は図 1(a) 上グラフの様になる。閾値とは、 $\Delta t = \Delta t_{th}$ となるようなデッドラインの確率値である。ここでは、 D_1 が閾値を下回るため、Value Function は図 1(b) 上グラフのように傾き 0 の直線となる。時間が経過し、同じタスクについて、 $\Delta t_1 > \Delta t_{th} > \Delta t_2$ となる D_2 が予測されると、デッドラインの確率分布において D_2 が閾値を上回る(図 1(a) 中グラフ)ため、Value Function はデッドライン後、傾きを持つ直線となる(図 1(b) 中グラフ)。また、Value Function の定義域が D_1 から D_2 まで移動する。さらに時間が経過して、 $\Delta t_1 > \Delta t_{th} > \Delta t_2 > \Delta t_3$ となる D_3 が予測されると、定義域が D_2 から D_3 まで移動し、デッドラインの確率分布は図 1(a) 下グラフの様になり、Value Function の傾きが大きくなって基本形に近づく。

このように、Variable Value Function モデルでは、デッドラインの確率分布によって Value Function の傾きが変化し、予測したデッドラインの変化に応じて、Value Function の定義域が変化する。本モデルでは、同一のタスクについても、時間が経過して、デッドラインが確実になるにつれて優先度が上がる。このことから、デッドラインが変化する本ケースでは、従来の Value Function モデルに比べ、Variable Value Function モデルのほうがより確実なユーザ待機時間短縮が期待できる。

4. まとめと今後の課題

本稿では、利用可能な場所が限定されているサービスについて、ユーザの移動状況の変化によって生じるサービス利用可能場所への到着時間予測の変化に対応したスケジューリングモデルを提案した。このスケジューリングモデルでは、予測した到着時刻をデッドラインとし、スケジューリング時刻とデッドラインとの時間差によって、Value Function の傾きを変化させた。またデッドライン予測が変更された場合は、Value Function の定義域を移動させた。

本稿では、単純に全ユーザの待機時間の総和を短縮させることを目的とした。そのため、Value Function のなかでも、一次関数の Linear Value Function [2] を適用した。しかし実世界では、少しならば待っていてもよい、などのユーザの嗜好を考慮してスケジューリングを行ったほうが、ユーザの不満は減少する。このようなスケジューリングを行うために、複雑な Quadratic Value Function [3] について本モデルを適用し、またユーザによって Value Function の形を自由に設定できるようにしたい。

参考文献

- [1] C.Mercer.: An Introduction to Real-Time Operating Systems: Scheduling Theory(1992)
- [2] R.McNaughton.: Scheduling with Deadlines and Loss Functions. *Management Science*, 6(1) pages 1-12(1959)
- [3] A.Schild and I.J.Fredman.: Scheduling Tasks with Deadlines and Non-linear Loss Functions. *Management Science*, 9(1) pages 73-81(1962)

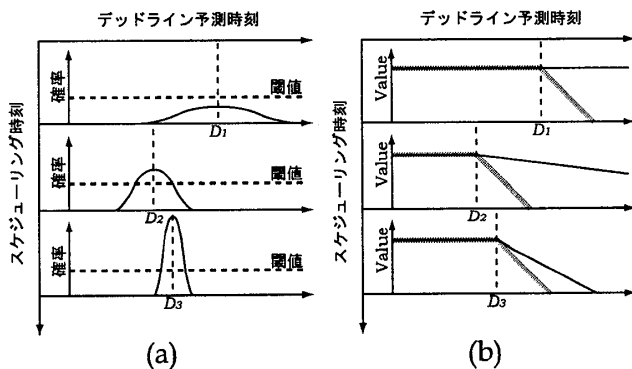


図 1: デッドライン確率分布と対応する Value Function

Function モデルでは、サービス側は予め二つの事柄を定める必要がある。一つめは、予測したデッドラインと実際の到着時間が等しい場合の、Value Function の基本形