Vol.2017-AL-164 No.4
2017/9/19

劣加法的集合関数の負荷分散最適化

永野 清仁[1,a)]　岸本 章宏[2,b)]

概要：劣モジュラ集合関数の最大化や最小化など，集合関数最適化の技術は昨今の人工知能や機械学習の分野において重要な役割を果たしている．劣加法的集合関数は，劣モジュラ性をシンプルに一般化した集合関数である．本研究では，劣加法的集合関数に関するミニマックスタイプの負荷分散最適化問題を扱い，モジュラ関数近似を用いた近似アルゴリズムを与える．また，集合関数が非負かつ劣加法的な場合について，このアルゴリズムの近似率を評価する．さらに，負荷分散最適化アルゴリズムをマルチロボット・ルーティング問題に適用し，アルゴリズムのパフォーマンスを計算機実験により評価する．

Subadditive Load Balancing

NAGANO, Kiyohito[1,a)]　KISHIMOTO, Akihiro[2,b)]

Abstract: Set function optimization such as submodular set function minimization and maximization is an essential technique in AI and machine learning. We focus on a subadditive set function that generalizes submodularity, and examine the subadditivity of some non-submodular functions. We also deal with a minimax subadditive load balancing problem, and present a modularization-minimization algorithm that theoretically guarantees a worst-case approximation factor for nondecreasing subadditive cases. We apply this approach to solve the multi-robot routing problem with the minimax team objective for an empirical performance evaluation.

1. はじめに

集合関数 (set function) は n 次元超立方体の頂点集合 $\{0,1\}^n$ の上で定義された離散領域上の関数とみなすことができる．特に，劣モジュラ集合関数 (submodular set function) は組合せ最適化における基本的な概念であると同時に，幅広い分野にわたる実際的な応用を持っている．例えば，劣モジュラ集合関数の最小化はクラスタリング [25], [27]，画像の領域分割 [14], [29] や特徴選択 [1], [2] など，機械学習の分野を中心とした多くの問題に適用されている．また，劣モジュラ関数最大化の典型的な応用な応用としては，影響最大化 [15]，センサ配置問題 [10]，文章要約 [23] などがある．

集合関数 f は，有限集合 $V=\{1,\ldots,n\}$ の部分集合全体の上で定義される実数値関数である．f の定義域は V のべき集合であり，これを $2^V=\{S:S\subseteq V\}$ と表記する．集合関数 $f:2^V\to\mathbb{R}$ が劣モジュラ (submodular) とは，任意の $S,T\subseteq V$ について，$f(S)+f(T)\geq f(S\cup T)+f(S\cap T)$ が成り立つことと定義される．劣モジュラ集合関数は，ある側面から見て，凸関数の離散版に対応する概念であることが知られている [24]．凸関数の場合と同様に，劣モジュラ関数は多項式時間で厳密に最小化することが可能である [9], [11], [28]．その一方で，実際的な状況でしばしば必要となるのだが，いくつかの単純な制約を追加した最適化問題にすることによって，劣モジュラ関数最小化は多くの場合難しい最適化問題になってしまう [6], [12], [30]．

本稿では負荷分散最適化として，ミニマックスタイプの問題を扱う．劣モジュラ負荷分散最適化問題は NP 困難な最適化問題であるが，Svitkina-Fleischer [30] はサンプリングベースの $O(\sqrt{n\ln n})$-近似アルゴリズムを与えている．Wei ら [33] は劣モジュラ負荷分散最適化問題を含む劣モジュラ分割問題について扱い，劣モジュラ負荷分散最適化問題に対してはモジュラ関数近似を用いた近似保証付きの近似アルゴリズムを与えている．彼らの近似率の解析は，劣モジュラ集合関数の曲率 [13], [32] の概念を用いている．

集合関数 $g:2^V\to\mathbb{R}$ が劣加法的 (subadditive) とは，任意の $S,T\subseteq V$ について，$g(S)+g(T)\geq g(S\cup T)$ が成り立つことと定義される．非負劣モジュラ性はただちに非負劣加法性を導く．非劣モジュラ最適化は機械学習の分野において重要になってきている [3]．しかし，我々の知る限りでは，劣モジュラ性のシンプルな一般化であるにも関わらず，一般の劣加法的最適化に対してはあまりにも少ない研究しか行われてきていない [5]．

本稿では，まず基本的な非劣モジュラ集合関数，施設配

[1] 群馬大学
Gunma University
[2] IBM Research, Ireland
[a)] nagano@si.gunma-u.ac.jp
[b)] akihirok@ie.ibm.com

置関数 [7] や最小全域木関数などの劣加法性について調べる．また，関数値情報が十分に与えられていないような劣モジュラ集合関数の劣加法的集合関数によるシンプルな補間手法についても示す．劣モジュラ関数の補間問題は劣モジュラ関数の近似問題 [8] と関連しているが，本アプローチはそれとはまったく異なるものである．また，Goemans ら [8] の手法は必ずしも実装が容易ではない．

そして本稿では，劣モジュラ負荷分散最適化の自然かつ重要な一般化として，ミニマックスタイプの劣加法的負荷分散最適化問題を扱う．本研究ではモジュラ関数近似を用いた近似アルゴリズムを与え，集合関数が非負かつ劣加法的な場合について，アルゴリズムの近似率を評価する．提案するアルゴリズムは，劣モジュラ負荷分散最適化に対する Wei らのアルゴリズム [33] の一般化に対応する．本研究での近似率の解析には劣加法的集合関数の曲率を用いている．目的関数がミニマックスタイプであるようなマルチロボット・ルーティング問題 [19] は最小全域木関数に関する劣加法的負荷分散最適化問題と深く関連している．マルチロボット・ルーティング問題に対して提案アルゴリズムを適用し，さらに既存のマルチロボット・ルーティングのアルゴリズムと比較することにより，アルゴリズムの実験的な性能について評価した．

本稿の構成は以下の通りである．2 節では劣加法的集合関数の例と劣加法的負荷分散問題の定義を与える．3 節では劣加法的負荷分散問題に対するアルゴリズムを記述し，近似率を解析する．4 節では劣加法的負荷分散問題とマルチロボット・ルーティング問題の関係について説明し，5 節では計算機実験の結果を示す．

2. 劣加法的負荷分散最適化問題

集合関数の基本概念と劣加法的集合関数の例について記述し，その後で劣加法的負荷分散最適化問題を定義する．

2.1 劣加法的集合関数

$V = \{1, \ldots, n\}$ を n 個の要素からなる有限集合，$g : 2^V \to \mathbb{R}$ を V の部分集合全体の上で定義された実数値関数とする．このような関数 g は，台集合を V とする集合関数とよばれる．集合関数 $g : 2^V \to \mathbb{R}$ は，$g(S) + g(T) \geq g(S \cup T), \forall S, T \subseteq V$ を満たすときに劣加法的，$g(S) + g(T) \geq g(S \cup T) + g(S \cap T), \forall S, T \subseteq V$ を満たすときに劣モジュラ，$g(S) + g(T) = g(S \cup T) + g(S \cap T), \forall S, T \subseteq V$ を満たすときにモジュラとよばれる．集合関数は $g(S) \geq 0, \forall S \subseteq V$ を満たすとき非負，$g(S) \leq g(T), \forall S, T \subseteq V$ with $S \subseteq T$ を満たすとき非減少とよばれ，$g(\varnothing) = 0$ を満たすとき標準化されているとよばれる．容易にわかるように，非負な劣モジュラ集合関数は非負な劣加法的集合関数である．よって，劣加法性は劣モジュラ性を単純に一般化しているといえる．

n 次元ベクトル $\boldsymbol{z} = (z_i)_{i \in V} \in \mathbb{R}^n$ と $S \subseteq V$ について，$z(S) = \sum_{i \in S} z_i$ と表記する．このようにして，ベクトル $\boldsymbol{z}$ に対応する集合関数 $z : 2^V \to \mathbb{R}$ が定まるが，z はモジュラ関数であり，$z(\varnothing) = 0$ を満たす．

2.2 劣加法的集合関数の例

非劣モジュラ集合関数の劣加法性について調べる．

2.2.1 劣モジュラ集合関数の劣加法的補間関数

$f : 2^V \to \mathbb{R}$ を，$f(\varnothing) = 0$ を満たす非減少な劣モジュラ関数とする．ここでは f の関数値情報の一部分しか与えられていないものと仮定する．つまり，ある集合族 $\mathcal{S} = \{S_1, S_2, \ldots, S_m\} \subseteq 2^V$ が与えられ，$f(S_i) = f_i \ (i = 1, \ldots, m)$ であることはわかっているが，$\mathcal{S}$ に属さない部分集合 $S \in 2^V \setminus \mathcal{S}$ については関数値 $f(S)$ がわからないとする．このとき，f をうまく近似するような集合 $g : 2^V \to \mathbb{R}$ を構成することを考える．ここでは，劣加法的な補間関数 $g_{\mathcal{S}}$ を構成する一般的かつシンプルな方法を提案する．この手法はポリマトロイド [4] と Lovász 拡張 [24] の考え方を利用する．

実際の応用において，f の関数値の計算量が大きくなるような場合が知られている ([22] など)．このような場合，$\mathcal{S}$ を適切に設定して提案する補間手法を用いることで，複雑な劣モジュラ最適化問題を単純な劣加法的最適化問題に置き換えることができる可能性がある．

Lovász 拡張

ポリマトロイド $\mathrm{P}(f) = \{\boldsymbol{z} \in \mathbb{R}^n : z(S) \leq f(S) \ (\forall S \subseteq V)\} \cap \mathbb{R}^n_{\geq 0}$ は有界な多面体である．ここで，$\mathbb{R}_{\geq 0}$ は非負実数全体の集合を表す．Lovász 拡張 $\widehat{f} : \mathbb{R}^n_{\geq 0} \to \mathbb{R}$ は $\widehat{f}(\boldsymbol{x}) = \max_{\boldsymbol{z} \in \mathrm{P}(f)} \langle \boldsymbol{x}, \boldsymbol{z} \rangle \ \ (\forall \boldsymbol{x} \in \mathbb{R}^n_{\geq 0})$ と定義される．ここで，$\langle \boldsymbol{x}, \boldsymbol{z} \rangle = \sum_{i \in V} x_i z_i$ とする．関数 $\widehat{f}$ について，$\widehat{f}(\boldsymbol{I}_S) = f(S), \forall S \subseteq V$ が成り立つので，$\widehat{f}$ は f の自然な連続拡張であるといえる．ここで $\boldsymbol{I}_S \in \{0, 1\}^n$ は S の特性ベクトルとする．

補間関数の構成方法

模倣ポリマトロイド $\mathrm{P}_{\mathcal{S}}(f)$ を $\mathrm{P}_{\mathcal{S}}(f) = \{\boldsymbol{z} \in \mathbb{R}^n : z(S_i) \leq f(S_i) \ (\forall i = 1, \ldots, m)\} \cap \mathbb{R}^n_{\geq 0}$，模倣 Lovász 拡張 $\widehat{f}_{\mathcal{S}}$ を $\widehat{f}_{\mathcal{S}}(\boldsymbol{x}) = \max_{\boldsymbol{z} \in \mathrm{P}_{\mathcal{S}}(f)} \langle \boldsymbol{x}, \boldsymbol{z} \rangle \ \ (\forall \boldsymbol{x} \in \mathbb{R}^n_{\geq 0})$ と定義する．$\widehat{f}_{\mathcal{S}}$ を用いて集合関数 $g_{\mathcal{S}} : 2^V \to \mathbb{R}$ を $g_{\mathcal{S}}(S) = \widehat{f}_{\mathcal{S}}(\boldsymbol{I}_S) \ (\forall S \subseteq V)$ と定義する．以下の補題は $g_{\mathcal{S}}$ が f の自然な劣加法的拡張であり，$g_{\mathcal{S}}$ が計算量的に扱い易いことを示している．

補題 1. $g_{\mathcal{S}} : 2^V \to \mathbb{R}$ は次の (i) –(iv) を満たす：(i) $g_{\mathcal{S}}(S) \geq f(S), \forall S \subseteq V$，(ii) $g_{\mathcal{S}}(S_i) = f(S_i), \forall i = 1, \ldots, m$，(iii) $g_{\mathcal{S}}$ は非減少な劣加法的集合関数，(iv) 任意の $S \subseteq V$ について関数値 $g_{\mathcal{S}}(S)$ は n と m の多項式時間で計算可能．

補題 1 (iii) の証明. $g_{\mathcal{S}}$ の非減少性は $\mathrm{P}_{\mathcal{S}}(f) \subseteq \mathbb{R}^n_{\geq 0}$ であることから導かれる．また，任意の $S, T \subseteq V$ に対し，$g_{\mathcal{S}}(S \cup$

$T) = \max_{\boldsymbol{z}\in \mathrm{P}_{\mathcal{S}}(f)}\langle \boldsymbol{I}_{S\cup T}, \boldsymbol{z}\rangle \leq \max_{\boldsymbol{z}\in \mathrm{P}_{\mathcal{S}}(f)}\langle \boldsymbol{I}_S + \boldsymbol{I}_T, \boldsymbol{z}\rangle \leq \max_{\boldsymbol{z}\in \mathrm{P}_{\mathcal{S}}(f)}\langle \boldsymbol{I}_S, \boldsymbol{z}\rangle + \max_{\boldsymbol{z}\in \mathrm{P}_{\mathcal{S}}(f)}\langle \boldsymbol{I}_T, \boldsymbol{z}\rangle = g(S) + g(T)$ が成り立つ．よって，$g_{\mathcal{S}}$ は劣加法的である． □

関数 $g_{\mathcal{S}}$ は必ずしも劣モジュラではない．例えば，$V = \{1,2,3\}$，$f(S) = (7 - |S|)|S|$ $(\forall S \subseteq V)$ とし，$\mathcal{S} = 2^V \setminus V$ とする．このとき，$g_{\mathcal{S}}(\{1\}) = 6$, $g_{\mathcal{S}}(\{1,2\}) = g_{\mathcal{S}}(\{1,3\}) = 10$, $g_{\mathcal{S}}(\{1,2,3\}) = 15$ より，$g_{\mathcal{S}}$ は劣モジュラではない．

2.2.2 最小全域木関数

最小全域木関数は劣加法的集合関数の標準的な例といえる．頂点 r をルートとし，$V = \{1, \ldots, n\}$ をその他の頂点の集合とする．任意の $i, j \in \widetilde{V} := \{r\} \cup V$ に対し，距離 $d(i,j) \geq 0$ が与えられているとする．ここで，$d : \widetilde{V} \times \widetilde{V} \to \mathbb{R}$ は対称的であり三角不等式を満たすものと仮定する．任意の部分集合 $S \subseteq V$ に対し，$\widetilde{S} := \{r\} \cup S$ に関する最小全域木 (minimum spanning tree) とは，$\widetilde{S}$ に関する全域木の中で枝の距離和を最小にするものと定義される．$S \subseteq V$ について，$MST(S)$ を $\widetilde{S}$ に関する最小全域木の枝の距離和と定める．集合関数 $MST : 2^V \to \mathbb{R}$ を最小全域木関数とよぶ．

補題 2. $MST : 2^V \to \mathbb{R}$ は非負かつ劣加法的である．

証明. 定義より，非負性は明らかである．$S, T \subseteq V$ に対し，E_S を $\widetilde{S}$ に関する最小全域木の枝集合，E_T を $\widetilde{T}$ に関する最小全域木の枝集合とする．このとき，頂点集合を $S \cup T \cup \{r\}$，枝集合を $E_S \cup E_T$ とするグラフ $(S \cup T \cup \{r\}, E_S \cup E_T)$ は連結である．よって，$MST(S) + MST(T) = \sum_{e\in E_S \cup E_T} d(e) \geq MST(S \cup T)$ が成り立ち，MST の劣加法性が示された． □

関数 $MST : 2^V \to \mathbb{R}$ について，非減少性や劣モジュラ性は必ずしも成り立たない．図 1 (a) の場合の MST について，$MST(\{1,3\}) = 10$，$MST(\{1,2,3\}) = 9$ となるので，MST は非減少ではない．図 1 (b) の場合の MST について，$MST(\{1\}) = 5$, $MST(\{1,2\}) = MST(\{1,3\}) = 6$, $MST(\{1,2,3\}) = 9$ が成り立つので，MST は劣モジュラではない．

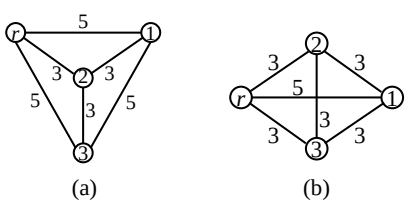

図 1 最小全域木関数

2.2.3 施設配置関数

施設配置関数 [7] もまた非劣モジュラであるような劣加法的集合関数の例になっている．$V = \{1, 2, \ldots, n\}$ を顧客の集合，F を施設の開設が可能な場所の有限集合とし，開設した施設は顧客になんらかのサービスを提供するものとする．施設 $j \in F$ の開設にはコスト $o_j \geq 0$ がかかり，顧客 $i \in V$ を施設 $j \in F$ につなげるにはコスト $c_{ij} \geq 0$ がかかる．顧客の部分集合 $S \subseteq V$ に対し，$FL(S)$ は S にサービスを提供するのに必要な最小のコストと定義する．このとき，$FL : 2^V \to \mathbb{R}$ を施設配置関数とよぶことにする．FL は次の性質を満たす．

補題 3. $FL : 2^V \to \mathbb{R}$ は非減少かつ劣加法的である．

すでに指摘されているように [7]，関数 $FL : 2^V \to \mathbb{R}$ は必ずしも劣モジュラではない．図 2 の場合の FL において，$F = \{a, b\}$，$V = \{1,2,3\}$ となるが，$FL(\{2\}) = 2$, $FL(\{1,2\}) = FL(\{2,3\}) = 3$, $FL(\{1,2,3\}) = 5$ が成り立つ．よって FL は劣モジュラではない．

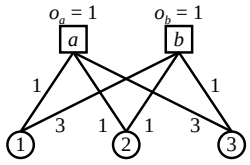

図 2 施設配置関数

2.3 劣加法的負荷分散最適化の定義

劣モジュラ負荷分散最適化 (submodular load balancing, 以下 SMLB と略す) 問題と，劣加法的負荷分散最適化 (subadditive load balancing, 以下 SALB と略す) 問題を定義する．$\mathcal{S} = (S_1, \ldots, S_m)$ が $V = \{1, \ldots, n\}$ の m 分割であるとは，$S_1 \cup \cdots \cup S_m = V$ かつ $S_i \cap S_j = \varnothing$ $(1 \leq i < j \leq m)$ (ここでは，ある S_j が空集合であっても構わない) を満たすことと定義する．$f_1, \ldots, f_m : 2^V \to \mathbb{R}$ を標準化された非負な劣モジュラ集合関数とし，$g_1, \ldots, g_m : 2^V \to \mathbb{R}$ を標準化された非負な劣加法的集合関数とする．SMLB 問題を次式で定義する．

$$\begin{aligned} \min \quad & \max_{j=1,\ldots,m} f_j(S_j) \\ \text{s.t.} \quad & \mathcal{S} = (S_1, \ldots, S_m) \text{ は } V \text{ の } m \text{ 分割.} \end{aligned} \tag{1}$$

$f_1, \ldots, f_m$ がすべて非減少であるとき，SMLB 問題もまた非減少であるという，非減少な SMLB 問題に対する近似アルゴリズムやヒューリスティクスが先行研究で与えられている [30], [33].

本稿で主に扱うのは，これまで十分に解析されていなかった，次式で定義される SALB 問題である．

$$\begin{aligned} \min \quad & \max_{j=1,\ldots,m} g_j(S_j) \\ \text{s.t.} \quad & \mathcal{S} = (S_1, \ldots, S_m) \text{ は } V \text{ の } m \text{ 分割.} \end{aligned} \tag{2}$$

SMLB の場合と同様に，$g_1, \ldots, g_m$ がすべて非減少であるとき，SALB 問題も非減少であるという．

3. 負荷分散最適化アルゴリズム

SMLB 問題の難しさから，SALB 問題もまた NP 困難である．よって，SALB 問題の近似解を得るアプローチを (理論保証がないものも含め) いくつか考える．以下に記述するアルゴリズム Greedy は単純な貪欲法である．

アルゴリズム Greedy$(g_1, \ldots, g_m)$

0: $S_j := \varnothing, \forall j \in [m] := \{1, \ldots, m\}, \quad U := V.$
1: **While** $U \neq \varnothing$ **do**
$\forall j \in [m]$ について $i_j \in \arg\min_{i \in U} g_j(S_j \cup \{i\})$ を選ぶ.
$j^* \in \arg\min_{j \in [m]} g_j(S_j \cup \{i_j\})$ を選ぶ.
$S_{j^*} := S_{j^*} \cup \{i_{j^*}\}, \quad U := U \setminus \{i_{j^*}\}.$
2: $\mathcal{S} = (S_1, \ldots, S_m)$ を出力する.

アルゴリズム Greedy はわかりやすいが，実験的にはあまり良い解を出力しない (5 節の実験結果を参照せよ).

劣モジュラの場合の SMLB 問題に対する自明でないアプローチとして，Wei ら [33] はモジュラ関数近似を用いた手法を提案し，問題が非減少である場合について近似率を与えている．この手法を拡張し，本研究では劣加法的な場合の SALB 問題に対し，モジュラ関数による近似を用いたアルゴリズムを提案する．劣モジュラ集合関数の場合と異なり，劣加法的集合関数は良い離散凸構造を持つわけではない．このため，劣モジュラの場合ではうまくいったことも，劣加法的な場合にはうまくいかない部分がある．ただし，本研究では問題が非減少な場合について，SMLB と同様の近似率を SALB でも達成可能であることを示す．本研究の解析では，劣加法的集合関数の曲率を用いる．

最小化問題 (P) とそれに対する近似アルゴリズム $\mathcal{A}$ について，$\alpha \geq 1$ として，$(OPT \leq)\ APP \leq \alpha \cdot OPT$ が常に成り立つとする．ここで OPT は問題 (P) の最適値，APP は $\mathcal{A}$ が出力する近似解に関する目的関数値としている．このとき，$\mathcal{A}$ は近似率 α を達成する，または，$\mathcal{A}$ は α-近似アルゴリズムであるという．

3.1 モジュラ関数近似を用いたアルゴリズム

SALB 問題に対する，モジュラ関数近似を用いたアルゴリズム MMin を記述する．

アルゴリズムの枠組み

アルゴリズム MMin は反復的に V の m 分割 $\mathcal{S}$ を更新する．各反復の操作について説明する．SALB 問題の暫定的な解として，m 分割 $\mathcal{S}' = (S'_1, \ldots, S'_m)$ が与えられているとする．このとき，各 $j = 1, \ldots, m$ について，関数 g_j を S'_j の周辺で近似するようなモジュラ近似関数 $M_j : 2^V \to \mathbb{R}$ を構成し，それらの近似関数を用いて新しい m 分割 $\mathcal{S}'' = (S''_1, \ldots, S''_m)$ を計算する．ここで，$\mathcal{S}''$ は次のモジュラ負荷分散最適化 (M-LB) 問題の最適解あるいは近似解とする:

$$\begin{aligned} \min \quad & \max_{j=1,\ldots,m} M_j(S_j) \\ \text{s.t.} \quad & \mathcal{S} = (S_1, \ldots, S_m) \text{ は } V \text{ の } m \text{ 分割.} \end{aligned} \tag{3}$$

劣加法的集合関数 $g : 2^V \to \mathbb{R}$ と部分集合 $S' \subseteq V$ が与えられたとき，関数 g を S' の周辺で近似するモジュラ関数として，例えば次式で定義されるモジュラ関数 $M : 2^V \to \mathbb{R}$ を用いることができる．

$$\begin{aligned} M(S) = g(S') &+ \sum_{i \in S \setminus S'} g(i \,|\, S') \\ &- \sum_{i \in S' \setminus S} g(i \,|\, S' \setminus \{i\}) \quad (S \subseteq V). \end{aligned} \tag{4}$$

ここで $S \subseteq V$ と $i \notin S$ に対し，$g(i \,|\, S) = g(S \cup \{i\}) - g(S)$ と表記している．もし g が非減少であれば，任意の $S \subseteq V$ と $i \notin S$ について $g(i \,|\, S)$ も非負となる．

アルゴリズムの記述

SALB 問題に対するモジュラ関数近似を用いたアルゴリズム MMin は以下のように記述される．

アルゴリズム MMin$(g_1, \ldots, g_m)$

0: 初期解となる V の m 分割 $\mathcal{S}^{(0)} = (S_1^{(0)}, \ldots, S_m^{(0)})$ を求め，$k := 1$ とおく.
1: 各 $j \in [m] = \{1, \ldots, m\}$ について，g_j の $S_j^{(k-1)}$ 周辺のモジュラ近似関数 $M_j^{(k)}$ を構成する.
2: $\max_{j=1,\ldots,m} M_j^{(k)}(S_j)$ の値を最小化するような V の m 分割 $\mathcal{S}$ を $\mathcal{S}^{(k)} = (S_1^{(k)}, \ldots, S_m^{(k)})$ とする.
3: **If** $\mathcal{S}^{(k)} = \mathcal{S}^{(k-1)}$
then $\mathcal{S} := \mathcal{S}^{(k)}$ を出力,
else $k := k + 1$ としてステップ 1 へ.

MMin では，ステップ 0 で初期解を必要とし，ステップ 2 で問題 M-LB (3) を解く必要がある．以下，これらをどのようにして実行するかを解説していく．

初期解を求める

アルゴリズム MMin は任意の初期解から開始することが可能である．ただ，非減少な場合の SALB 問題で近似率を評価するためには，モジュラ負荷分散問題

$$\begin{aligned} \min \quad & \max_{j=1,\ldots,m} M_j^{(0)}(S_j) \\ \text{s.t.} \quad & \mathcal{S} = (S_1, \ldots, S_m) \text{ は } V \text{ の } m \text{ 分割,} \\ & \text{where } M_j^{(0)}(S) = \sum_{i \in S} g_j(\{i\}),\ \forall j \in [m] \end{aligned} \tag{5}$$

を考え，この最適解 $\mathcal{S}^{(0)}$ を初期解とする．

モジュラ負荷分散問題を解く

問題 (3) における各モジュラ関数 M_j は，$M_j(S) =$

$b_j + \sum_{i\in S} c_{ij}$ $(S \subseteq V)$ の形で表現することが可能である．よって，標準的な IP (integer programming) の定式化手法を用いて，問題 (3) は次の形に変形される．

$$\begin{aligned} \min \quad & y \\ \text{s.t.} \quad & \sum_{j\in[m]} x_{ij} = 1,\ \forall i \in V, \\ & b_j + \sum_{i\in S} c_{ij}x_{ij} \le y,\ \forall j \in [m], \\ & x_{ij} \in \{0,1\}, \forall i \in V,\ \forall j \in [m],\quad y \in \mathbb{R}. \end{aligned} \tag{6}$$

問題 (6) の最適解は，IBM ILOG CPLEX のような IP ソルバを用いて求めることができる．あるいは，無関連並列マシンスケジューリング問題に対する LP ベースの 2-近似アルゴリズム [21] を用いて問題 (6) を解くこともできる．ただし，2-近似アルゴリズムの近似率 2 が保持されるのは，$b_j \ge 0$ $(\forall j \in [m])$ かつ $c_{ij} \ge 0$ $(\forall i \in V, \forall j \in [m])$ が成り立つ場合のみである．

3.2 近似アルゴリズムの解析

非減少な SALB 問題に対し，アルゴリズム MMin の近似率を評価する．次の結果は非減少な SMLB 問題に対する [33] の結果の一般化である．

定理 4. 非減少な劣加法的負荷分散問題 (2) に対し，アルゴリズム MMin は近似率 $2 \cdot (\max_{j\in[m]} \frac{|S_j^*|}{1+(|S_j^*|-1)(1-\kappa_g(S_j^*))})$ を達成する．ここで $\mathcal{S}^* = (S_1^*, \ldots, S_m^*)$ は問題 (2) の最適解であり，$\kappa_{g_j}(S)$ は g_j の $S \subseteq V$ における曲率である．

定理 4 を証明するために，劣加法的集合関数の曲率を定義し，その性質について調べる．

曲率とモジュラ近似関数

劣モジュラ関数の場合 [13], [32] と同様に，劣加法的集合関数の曲率を定める．ここで関数 g は標準化された非減少な劣加法的集合関数であり，各 $i \in V$ について $g(\{i\}) > 0$ が成り立つことを仮定する．このとき，関数 g の $S \subseteq V$ における曲率 $\kappa_g(S)$ を次式で定義する．

$$\kappa_g(S) = 1 - \min_{A\subseteq S,\, i\in A} \frac{g(i \,|\, A \setminus \{i\})}{g(\{i\})}.$$

$\kappa_g(V)$ を全曲率とよび，κ_g と表記する．定義より，任意の $S' \subseteq S \subseteq V$ と $i \in S'$ に対し，$\frac{g(i\,|\,S'\setminus\{i\})}{g(\{i\})} \ge 1 - \kappa_g(S)$ が成り立つので，次式が得られる．

$$g(i \,|\, S' \setminus \{i\}) \ge (1 - \kappa_g(S))g(\{i\}). \tag{7}$$

集合関数 $\widehat{g} : 2^V \to \mathbb{R}$ が g の α-近似とは，任意の $S \subseteq V$ に対し，$g(S) \le \widehat{g}(S) \le \alpha g(S)$ が成り立つことと定義する．次の補題はモジュラ関数 $M(S) = \sum_{i\in S} g(\{i\})$ $(S \subseteq V)$ がどの程度 g を近似するかを評価するものである．

補題 5. $0 < \kappa_g < 1$ のとき，$g(S) \le \sum_{i\in S} g(\{i\}) \le \frac{1}{1-\kappa_g(S)} g(S)$ $(S \subseteq V)$ が成り立つ．

証明. 1 つ目の不等式は劣加法性から直ちに成り立つ．$|S| = h$ とし，$S = \{i_1, \ldots, i_h\}$ とおく．各 $k = 1, \ldots, h$ に対し，$S_k = \{i_1, \ldots, i_k\}$ と定める．このとき，不等式 (7) を用いて $g(S) = \sum_{k=1}^{h} g(i_k \,|\, S_k \setminus \{i_k\}) \ge (1 - \kappa_g) \sum_{i\in S} g(\{i\})$ が得られ，2 つ目の不等式も示された． □

さらに詳細な解析によって，任意の $S \subseteq V$ に対し次式が得られる．

$$\sum_{i\in S} g(\{i\}) \le \frac{|S|}{1 + (|S| - 1)(1 - \kappa_g(S))} g(S). \tag{8}$$

近似率の解析

ここでは $g_1, \ldots, g_m$ は非減少とし，任意の $i \in V$ と $j \in [m]$ について，$g_j(\{i\}) > 0$ が成り立つものと仮定する．さらに，アルゴリズム MMin のステップ 0 で解く必要がある問題 (5) に対し，β-近似アルゴリズムを用いるものとする．例えば，Lenstra ら [21] の多項式時間アルゴリズムは問題 (5) に対して近似率 2 を達成する．

定理 4 を証明するには，アルゴリズム MMin の初期解 $\mathcal{S}^{(0)}$ ですでに示したい近似率を達成していることを証明する．つまり，以下の補題を証明すれば十分である．

補題 6. $\mathcal{S}^{(0)} = (S_1^{(0)}, \ldots, S_m^{(0)})$ を問題 (5) の最適解とし，$\mathcal{S}^* = (S_1^*, \ldots, S_m^*)$ を非減少な問題 SALB の最適解とする．このとき，$\mathcal{S}^{(0)}$ は非減少な問題 SALB の $(\max_{j\in[m]} \frac{|S_j^*|}{1+(|S_j^*|-1)(1-\kappa_g(S_j^*))})$-近似解である．

証明. 各 $j \in [m]$ について，$\alpha_j^* = \frac{|S_j^*|}{1+(|S_j^*|-1)(1-\kappa_g(S_j^*))}$ とする．不等式 (8) から，各 $j \in [m]$ について $\sum_{i\in S_j^*} g_j(\{i\}) \le \alpha_j^* g_j(S_j^*)$ が得られる．よって，次式が成立する．

$$\max_{j\in[m]} \sum_{i\in S_j^*} g_j(\{i\}) \le \left(\max_{j\in[m]} \alpha_j^*\right) \cdot \left(\max_{j\in[m]} g_j(S_j^*)\right). \tag{9}$$

また，劣加法性と $\mathcal{S}^{(0)}$ の最適性から次式が得られる．

$$\begin{aligned} \max_{j\in[m]} g_j(S_j^{(0)}) &\le \max_{j\in[m]} \sum_{i\in S_j^{(0)}} g_j(\{i\}) \\ &\le \max_{j\in[m]} \sum_{i\in S_j^*} g_j(\{i\}). \end{aligned} \tag{10}$$

不等式 (9) と (10) から，m 分割 $\mathcal{S}^{(0)}$ が非減少な SALB 問題の $(\max_{j\in[m]} \alpha_j^*)$-近似解であることがわかる． □

4. マルチロボット・ルーティングへの応用

この節では，劣加法的負荷分散最適化と目的関数がミニマックスタイプであるマルチロボット・ルーティング問題の関係について説明する．

$\mathcal{R} = \{r_1, \ldots, r_m\}$ をロボットの集合，$\mathcal{T} = \{t_1, \ldots, t_n\}$ をターゲットの集合とする．各 $i, j \in \mathcal{R} \cup \mathcal{T}$ に対し，非負のコスト (i と j の距離) $d(i, j) \ge 0$ が定まっており，コス

ト関数 $d : (\mathcal{R} \cup \mathcal{T}) \times (\mathcal{R} \cup \mathcal{T}) \to \mathbb{R}$ は対称的かつ三角不等式を満たすものとする．ここで，ターゲットのロボットへの割り当てを考え，$S_j \subseteq \mathcal{T}$ をロボット $r_j \in \mathcal{R}$ に割り当てられるターゲット部分集合とする．目的関数がミニマックスタイプであるマルチロボット・ルーティング (MRR) 問題とは，$\mathcal{T}$ の m 分割 $\mathcal{S} = (S_1, \ldots, S_m)$ と，各ロボット $r_j \in \mathcal{R}$ について r_j が S_j に属するターゲットすべて回収するようなパス P_j を決定する問題であり，以下のように表される最適化問題である：

$$\min_{\mathcal{S}} \max_{j \in \mathcal{R}} RPC_j(S_j) \quad \text{または} \quad \min_{\mathcal{S}} \max_{j \in \mathcal{R}} RTC_j(S_j).$$

ここで，$RPC_j(S_j)$ はロボット $r_j \in \mathcal{R}$ が S_j のターゲットすべてを回収するようなパス P_j のコストの最小値 (ロボットパスコスト, RPC) を表し，$RTC_j(S_j)$ は頂点部分集合 $\{r_j\} \cup S_j$ 上のコスト関数 d に関する最小全域木のコスト (ロボットツリーコスト, RTC) を表す．パスは木 (ツリー) の一種なので，$RPC_j(S_j) \geq RTC_j(S_j)$ が成り立つ．

巡回セールスマン問題 (TSP) の難しさから，RPC は計算量的に扱い易くない．その一方で，RTC は扱い易く，さらに $\{r_j\} \cup S_j$ 上の最小全域木は $\{r_j\} \cup S_j$ 上のロボットパスでコストが $1.5 \cdot RPC_j(S_j)$ 以下のものに比較的容易に変換可能である (例えば [31] などを参照されたい)．RPC_j の近似として RTC_j は妥当な関数といえる．

MRR 問題に対する従来手法として，特に SSI (Sequential Single-Item) オークションに基づいた近似アルゴリズムが広く研究されている [17]．このアプローチでは割り当て問題をオークションとしてとらえ，ロボットを入札者，ターゲットを商品とそれぞれみなす．オークションは，複数回 (ターゲットの個数回) のラウンドから構成される．どのターゲットもロボットに割り当てられていない状態から開始し，ラウンド 1 回ごとにただ 1 つのターゲットの割り当てが決まる．各ラウンドの流れを説明する．各ロボットは，現在そのロボットに割り当てられているターゲット集合とまだ割り当てられていないターゲットに関する距離情報を用いて，入札するターゲット 1 つと入札値を決定する．すべてのロボットの入札値を受け，勝者となるロボットがただ 1 つ決定され，対応する 1 つのターゲットがオークションの勝者に割り当てられる．以上の手続きは，すべてのターゲットが割り当てられるまで行われる．

劣加法的負荷分散最適化手法を用いることで，MRR 問題に対する新しいアプローチを与えることができる．これは関数 $RTC_j(S_j)$ が最小全域木関数 (§2.2) なので，劣加法的集合関数となるためである．

5. 計算機実験

マルチロボット・ルーティング (MRR) 問題に対するアルゴリズムの性能評価のために，Intel Core i5-6300U (クロック数 2.40GHz，4CPU コア) と 8GB のメモリで構成される PC 上で，1 コアのみを利用して実験を行った．本実験では，3 節のアルゴリズム MMin を C++で実装し，MMin の中で現れる部分問題 (6) は IBM ILOG CPLEX を用いて解いた．また，以下の代表的な MRR アルゴリズムを実装した：

- 最小全域木に基づいたアルゴリズム MST [19] は本稿のアルゴリズム Greedy と一致する [*1]．本実験では性能評価のために，RTC と RPC の両方の値を用いる．MST/MMin において RPC の値を計算する際に，各ロボットの持つツリー (最小全域木) をパスに置き換える必要がある．ここでは，MRR ではよく行われるように [16], [19]，ショートカット法 [20] によってパスを生成している．
- Path [17], [19] はオークションに基づく標準的なアルゴリズムである．各ロボットは，枝を何も持っていない状況から開始し，TSP の挿入ヒューリスティクス [20] を用いて，貪欲にパスを拡大していく．オークションの各ラウンドでは，例えば最小 (あるいは準最適) な RPC を達成するロボットが割り当てられていないターゲットを獲得する．以上の手順はすべてのターゲットが割り当てられるまで行われる．Path は直接 RPC の値を計算するため，(RTC は用いることができないので) アルゴリズムの性能比較では RPC を基準として用いる．

本実験では，日本の函館地区のロードマップを用意し，さらに 2 点間の距離については Open Source Routing Machine[*2] を用いて前計算を行った．このようにして本実験では全距離情報を持つことで距離情報を直ちに得られるが，このような状況は MRR 問題の一般的な問題設定の一つである [18], [34]．実際，函館のような小さい都市にマップを限定するときは，全距離情報を利用して乗り合いタクシーの経路計算などが可能である [26]．

本実験では，ロボット数を 5 に固定し，ターゲット数は 50 と 100 の 2 つのケースを扱った．マップ上にロボットとターゲットをランダムに配置することで，各ケースについて 100 個のインスタンスを生成した．

表 1 は RTC の値と計算時間の平均値を示している．ここで Path が含まれていないのは，Path が RTC の計算を用いないアルゴリズムであるためである．また，表 1 の「初期分割」の値は，MMin の初期解である m 分割として 3 節で述べたものについて，その RTC の値と，それを見つけるのにかかった計算時間を表している．反復アルゴリズムである MMin は，初期分割から解を改善していく．初期分割としては 3 節で述べたものと異なる m 分割を用いることも可能である．MMin + MST は，初期分割として

[*1] Wei ら [33] はこのアルゴリズムと非常に近いアルゴリズム GreedyMin を用いている．
[*2] http://project-osrm.org/

表 1 MST と MMin の RTC の値と計算時間の比較

	MST		初期分割		MMin		MMin + MST	
ターゲット数	RTC	時間 (s)	RTC	時間 (s)	RTC	時間 (s)	RTC	時間 (s)
50	42,208	0.0002	47,392	0.80	38,202	1.29	36,326	0.63
100	54,840	0.17	66,131	12.89	50,437	13.81	47,752	0.76

表 2 各手法の RPC の値の比較

ターゲット数	MST	Path	MMin	MMin + MST
50	52,336	**49,792**	50,984	50,446
100	72,553	69,002	68,282	**68,067**

MST により求めた m 分割を用いてアルゴリズム MMin を実行したものである．

RTC の値について，MMin は MST と比べ，ターゲット数 50 の場合は 10%，ターゲット数 100 の場合は 8%小さい値の解をそれぞれ生成している．MMin の初期分割の状態では MST よりも劣っているが，MMin の反復操作によって，初期分割がうまく改善することがわかる．

ターゲット数 50 のすべてのインスタンスについて, MMin の反復回数は平均 17.92 回 (最小 4 回, 最大 56 回) であった．ターゲット数 100 の場合は，平均反復回数は 21.45 回 (最小 5 回, 最大 62 回) であった．MMin が多くの反復回数を要する場合であったとしても，初期分割を求めるのにかかる計算時間がかなりの割合を占め，全体の計算時間の 62-93 %となった．ソルバーである CPLEX にとって，初期分割を求めるための IP は，それ以降の MMin のすべての反復で現れる IP よりも難しいということがわかる．初期分割の計算時間は，ターゲット数 50 の場合 0.80 秒，ターゲット数 100 の場合 12.89 秒となり，ターゲット数の増加によりかなり難しい計算となることもわかる．ターゲット数 50 で最も計算時間のかかったインスタンスでは，MMin は初期分割を求めるのに 19.67 秒を必要とし，残りのすべての反復の計算は合計でたった 0.72 秒しかかからなかった．ターゲット数 100 の最悪ケースのインスタンスでは，初期分割の計算に 365.87 秒，残りの計算に 4.53 秒がそれぞれかかった．

RTC の値と計算時間の両方について，MMin + MST は最も良い値を示している．これは, MMin においてオーバーヘッドとなった初期分割の計算を避けられているだけではなく，MMin の初期解よりも良い MST の解を初期解として用いているためだと考えられる．このことは，実用上はこのようなハイブリッドなアプローチが重要であることを示している．オークションに基づいた MRR アルゴリズムについて，[19] ではその近似率の限界に関する議論している．しかし，アルゴリズム MMin は分散型ではなく集中型のアプローチである．MMin + MST のような集中型のアルゴリズムによって，MRR 問題に対し既存の分散型アルゴリズムやその限界値よりも良い近似率を達成可能かどうかの検証は，今後の重要な課題である．

表 2 は各手法の RPC の値を示している．ツリーをパスに置き換えるショートカット法は近似アルゴリズムなので，理論上は RTC の値が良いからといって，RPC の値も良いとは限らない．しかし，実際は RPC を基準としても，MMin は MST よりも良い性能を示している．RPC の値の平均値について，MMin は MST と比較して，ターゲット数 50 の場合は 3%，ターゲット数 100 の場合は 6%良い解を出力している．

RPC について，ターゲット数 50 の場合は Path が最も良い値を示しているが，ターゲット数 100 の場合は MMin の方が Path よりも良い．また，MMin と MST を組み合わせた手法は，RPC の値について MMin を若干上回り，ターゲット数 100 の場合は最も良い値を示している．

MRR 問題において，最適な RPC の値と比較して，Path と MST は同じ近似率を達成することが知られているが，Path の方が MST よりも良い性能を示す傾向にあるという共通の認識がある [19]．本実験においても MST と Path の単純比較ではその傾向は正しい．しかし，本実験結果は，RTC を基準として (つまり最小全域木関数を用いて) 求めたツリーの解をパスに置き換えるというアプローチが Path の性能を超えるポテンシャルを持つこと示しており，MRR 問題に対するさらなる研究の可能性を示唆している．

6. まとめ

本研究では，劣モジュラ負荷分散最適化の一般化である劣加法的負荷分散最適化問題に対し，モジュラ関数近似を用いたアルゴリズムを提案し，劣加法的関数が非減少な場合の近似率を評価した．さらに，マルチロボット・ルーティングへの応用を通じてアルゴリズムの性能を評価した．人工知能分野への劣加法的集合関数の応用は新しいものであり，劣加法的最適化によるアプローチは人工知能や機械学習分野の新たな方向性を示唆するものである．今後の課題として，初期解を変えたときの提案アルゴリズムの理論的および実験的性能に関するさらなる詳細な解析が挙げられる．また，提案アルゴリズムの反復操作が近似率の理論的な評価を改善するかどうかも明らかにはなっていない．

参考文献

[1] Bach, F.: Structured sparsity-inducing norms through submodular functions, *NIPS*, pp. 118–126 (2010).
[2] Bach, F.: Learning with Submodular Functions: A Convex Optimization Perspective, *Foundations and Trends in Machine Learning*, Vol. 6, No. 2–3, pp. 145–373 (2013).
[3] Bian, A. A., Buhmann, J. M., Krause, A. and Tschiatschek, S.: Guarantees for Greedy Maximization of Non-submodular Functions with Applications, *CoRR*, Vol. abs/1703.02100 (online), available from ⟨http://arxiv.org/abs/1703.02100⟩ (2017).
[4] Edmonds, J.: Submodular functions, matroids, and certain polyhedra, *Combinatorial Structures and Their Applications* (Guy, R., Hanani, H., Sauer, N. and Schönheim, J., eds.), Gordon and Breach, pp. 69–87 (1970).
[5] Feige, U.: On Maximizing Welfare When Utility Functions Are Subadditive, *SIAM J. Comput.*, Vol. 39, No. 1, pp. 122–142 (2009).
[6] Goel, G., Karande, C., Tripathi, P. and Wang, L.: Approximability of combinatorial problems with multi-agent submodular cost functions, *FOCS'09*, pp. 755–764 (2009).
[7] Goemans, M. X. and Skutella, M.: Cooperative facility location games, *Journal of Algorithms*, Vol. 50, pp. 194–214 (2004).
[8] Goemans, M. X., Harvey, N. J. A., Iwata, S. and Mirrokni, V.: Approximating Submodular Functions Everywhere, *Proceedings of the Twentieth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '09, Philadelphia, PA, USA, Society for Industrial and Applied Mathematics, pp. 535–544 (online), available from ⟨http://dl.acm.org/citation.cfm?id=1496770.1496829⟩ (2009).
[9] Grötschel, M., Lovász, L. and Schrijver, A.: *Geometric Algorithms and Combinatorial Optimization*, Springer (1988).
[10] Guestrin, C., Krause, A. and Singh, A.: Near-Optimal Sensor Placements in Gaussian Processes, *ICML*, pp. 265–272 (2005).
[11] Iwata, S., Fleischer, L. and Fujishige, S.: A combinatorial strongly polynomial algorithm for minimizing submodular functions, *Journal of the ACM*, Vol. 48, pp. 761–777 (2001).
[12] Iwata, S. and Nagano, K.: Submodular function minimization under covering constraints, *FOCS'09*, pp. 671–680 (2009).
[13] Iyer, R. K., Jegelka, S. and Bilmes, J. A.: Curvature and Optimal Algorithms for Learning and Minimizing Submodular Functions, *NIPS*, pp. 2742–2750 (2013).
[14] Jegelka, S. and Bilmes, J.: Submodularity beyond submodular energies: coupling edges in graph cuts, *CVPR'11*, pp. 1897–1904 (2011).
[15] Kempe, D., Kleinberg, J. and Tardos, E.: Maximizing the spread of influence through a social network, *KDD'03*, pp. 137–146 (2003).
[16] Kishimoto, A. and Sturtevant, N.: Optimized Algorithms for Multi-Agent Routing, *AAMAS*, pp. 1585–1588 (2008).
[17] Koenig, S., Keskinocak, P. and Tovey, C.: Progress on Agent Coordination with Cooperative Auctions, *AAAI*, pp. 1713–1717 (2010).
[18] Koenig, S., Tovey, C., Zheng, X. and Sungur, I.: Sequential Bundle-Bid Single-Sale Auction Algorithms for Decentralized Control, *IJCAI*, pp. 1359–1365 (2007).
[19] Lagoudakis, M. G., Markakis, E., Kempe, D., Keskinocak, P., Kleywegt, A., Koenig, S., Tovey, C., Meyerson, A. and Jain, S.: Auction-Based Multi-Robot Routing, *Proc. of Robotics: Science and Systems* (2005).
[20] Lawler, E. L., Lenstra, J. K., Kan, A. H. G. R. and Shmoys, D. B.: *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization*, Wiley Series in Discrete Mathematics and Optimization (1985).
[21] Lenstra, J. K., Shmoys, D. B. and Tardos, E.: Approximation Algorithms for Scheduling Unrelated Parallel Machines, *Math. Program.*, Vol. 46, pp. 259–271 (1990).
[22] Leskovec, J., Krause, A., Guestrin, C., Faloutsos, C., VanBriesen, J. and Glance, N.: Cost-effective Outbreak Detection in Networks, *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '07, New York, NY, USA, ACM, pp. 420–429 (online), DOI: 10.1145/1281192.1281239 (2007).
[23] Lin, H. and Bilmes, J.: Multi-document summarization via budgeted maximization of submodular functions, *HLT'10*, pp. 912–920 (2010).
[24] Lovász, L.: Submodular functions and convexity, *Mathematical Programming — The State of the Art* (Bachem, A., Grötschel, M. and Korte, B., eds.), Springer-Verlag, pp. 235–257 (1983).
[25] Nagano, K., Kawahara, Y. and Iwata, S.: Minimum average cost clustering, *NIPS*, pp. 1759–1767 (2010).
[26] Nakashima, H., Sano, S., Hirata, K., Shiraishi, Y., Matsubara, H., Kanamori, R., Koshiba, H. and Noda, I.: One Cycle of Smart Access Vehicle Service Development, *Proc. of the 2nd International Conf. on Serviceology*, pp. 152–157 (2014).
[27] Narasimhan, M., Jojic, N. and Bilmes, J.: Q-clustering, *NIPS*, pp. 979–986 (2005).
[28] Schrijver, A.: A combinatorial algorithm minimizing submodular functions in strongly polynomial time, *Journal of Combinatorial Theory (B)*, Vol. 80, pp. 346–355 (2000).
[29] Stobbe, P. and Krause, A.: Efficient minimization of decomposable submodular functions, *NIPS*, pp. 2208–2216 (2010).
[30] Svitkina, Z. and Fleischer, L.: Submodular approximation: sampling-based algorithms and lower bounds, *FOCS'08*, pp. 697–706 (2008).
[31] Vazirani, V. V.: *Approximation Algorithms*, Springer-Verlag New York, Inc. (2001).
[32] Vondrak, J.: Submodularity and curvature: the optimal algorithm, *RIMS Kokyuroku Bessatsu, volume B23*, Vol. B23, pp. 253–266 (2010).
[33] Wei, K., Iyer, R. K., Wang, S., Bai, W. and Bilmes, J. A.: Mixed Robust/Average Submodular Partitioning: Fast Algorithms, Guarantees, and Applications, *NIPS*, pp. 2233–2241 (2015).
[34] Zheng, X., Koenig, S. and Tovey, C.: Improving Sequential Single-Item Auctions, *Proc. of the IEEE International Conf. on Intelligent Robots and Systems (IROS'06)*, pp. 2238– 2244 (2006).