RF-002

分散ラグランジュ緩和プロトコルにおける局所情報にもとづく価格更新の効果
Effects of price update based on local information in Distributed Lagrangian Relaxation Protocol

浅野大介 † Daisuke Asano　松井俊浩 † Toshihiro Matsui　松尾啓志 † Hiroshi Matsuo

1. はじめに

複数の自律的な主体 (エージェント) が協調して問題を解決していくマルチエージェントシステムは，人工知能分野の重要な課題として研究されている．スマートグリッドやセンサネットワークなどに含まれる，分散プランニングや分散スケジューリング問題，組み合わせオークション [1] の問題などは，複数エージェントに分散して配置された組み合わせ最適化問題として定式化できる場合がある．

このような組み合わせ最適化問題として，一般化割当問題 (Generalized Assignment Problem: GAP) [2] を分散環境に適応するように拡張した，一般化相互割当問題 (Generalized Mutual Assignment Problem: GMAP) が提案された．また，GMAP を解くアルゴリズムとして分散ラグランジュ緩和プロトコル (Distributed Lagrangian Relaxation Protocol: DisLRP) [2, 3, 4] が提案されている．

GMAP は，ジョブを持つ複数のエージェントが各々のジョブを系全体で相互に割り当てる (交換する) 際，各エージェントの容量制約を満たしつつ系全体の効用の和が最大となるような割り当ての組み合わせを求める分散最大化問題である．

DisLRP はエージェント間の局所的通信を用いて GMAP を解くアルゴリズムである．このアルゴリズムでは，ラグランジュ緩和を用いて，GMAP を個々のエージェントに関連するジョブについての 0-1 ナップサック問題に分割する．そして，各エージェントは，自身がどのジョブを選択したかを通知する．「全てのジョブが 1 つのエージェントのみに選択されている状態」でない場合はジョブの価格を更新し，0-1 ナップサック問題を解く処理を繰り返す．

従来の DisLRP [2] ではジョブの効用値を更新する際，その更新幅を決定するステップ長に定数値のパラメータを用いていた．各エージェントに同様の規則を用いることにより分散環境における解法を比較的容易に構成でき，その値を適切に選べば，準最適な解に収束する．しかし，その適切な定数値は自明では無く，また問題例毎に異なるため，ユーザが問題ごとに手動で調整しなければならなかった．

これに対し，オンラインで系全体の大域情報を収集し，その情報を基にステップ長を適応的に決める更新規則を分散環境で実現した ADisLRP(Adaptive DisLRP) [5, 6] が提案された．この手法は系全体の解の上界値と下界値を大域情報から推定し，適応的にステップ長を更新するものである．しかしこの手法ではエージェントが個別に持つジョブの情報を他のエージェントに通知するための，追加的な通信が必要となる．また，エージェントの持つジョブの効用値情報が他のエージェントに伝搬されるため，各エージェントのプライバシーを保護する観点において課題がある [7].

そこで本論文では二つの手法を提案する．1 つめの手法では，問題毎に適切に設定するべきパラメータを，従来の DisLRP において個々のエージェントが知る情報のみから導出する．すなわち，エージェントが個別に持つジョブの情報と，関連するエージェントのジョブの決定変数から得られる平均値を用いる．2 つめの手法では，系内での解の収束の状況，すなわち，割り当てに関する制約に違反している数に応じてパラメータを動的に調整する．

提案手法により問題に応じたパラメータを設定することにより，局所的な情報のみを用いても，大域的な情報を利用する従来手法に比較的近い，計算回数と解の精度が得られることが期待される．計算機シミュレーションによる評価に基づいて，提案手法の有効性を評価した．

以降の本論文は次のように構成される．まず，2 章で一般化相互割当問題について述べる．3 章では，従来手法の DisLRP および ADisLRP の概略を示す．4 章では，局所情報からパラメータを計算する手法を提案する．5 章では，提案手法を実験により評価する．最後に 6 章で本研究のまとめと今後の課題を示す．

2. 一般化相互割当問題

2.1. 一般化割当問題

GMAP [2] はジョブを持つ複数のエージェントが各々のジョブを相互に割り当てる (交換する) 際，各エージェントの資源制約を満たしつつ且つ系全体の効用値の和が最大となる割り当てを求める分散最大化問題である．系全体で見た場合，GMAP におけるエージェントは次の様に定式化された整数計画問題 $\mathcal{GAP}$ を解く．

$$\mathcal{GAP}\ (\text{decide } x_{ij},\ \forall i \in A,\ \forall j \in J):$$

$$\max.\ \sum_{i \in A}\sum_{j \in J} p_{ij}x_{ij} \tag{1}$$

$$\text{s.t.}\ \sum_{i \in A} x_{ij} = 1,\ \forall j \in J, \tag{2}$$

$$\sum_{j \in J} \omega_{ij}x_{ij} \leq c_i,\ \forall i \in A, \tag{3}$$

$$x_{ij} \in \{0, 1\},\ \forall i \in A,\ \forall j \in J. \tag{4}$$

ここで，$A = \{1, \ldots, m\}$ はエージェントの集合，$J = \{1, \ldots, n\}$ はジョブの集合，p_{ij} はエージェント i がジョブ j を得た場合の効用値，ω_{ij} はエージェント i がジョ

†名古屋工業大学, Nagoya Institute of Technology

ブ j を得た場合の資源消費量，c_i はエージェント i の資源容量である．また，x_{ij} は，エージェント i にジョブ j が割り当てられた場合は 1，そうでない場合は 0 に設定される決定変数である．問題の目的は各ジョブがただ一つのエージェントに割り当てられ (割当制約, 式 (2))，どのエージェントもその資源消費量の合計が資源容量を超えない (ナップサック制約， 式 (3))，かつ，どのジョブもエージェントに割り当てられるか割り当てられないかのどちらかである (01 制約， 式 (4)) という制約条件のもとで効用値の総和を最大化する事である．この問題は NP 困難な問題に属すため，近似解法も研究されている．

2.2.GMAP への拡張

分散型問題の表現と解法を目指す場合は，大域的な定式化ではなく，個々のエージェントが全体の問題の内のどの部分を担当するかが重要となる．そこで前述した定式化における決定変数 x_{kj} の値はエージェント k が決定する，すなわちジョブの割り当て側では無く受け手側に決定権があるものとし，GMAP をエージェント k に対する整数計画問題

$$\mathcal{GMP}_k \text{ (decide } x_{kj},\ \forall j \in R_k) :$$

$$\text{max.} \sum_{j \in R_k} p_{kj} x_{kj} \tag{5}$$

$$\text{s.t.} \sum_{k \in S_j} x_{kj} = 1,\ \forall j \in R_k, \tag{6}$$

$$\sum_{j \in R_k} \omega_{kj} x_{kj} \leq c_k, \tag{7}$$

$$x_{kj} \in \{0,1\},\ \forall j \in R_k. \tag{8}$$

の集合 $\{\mathcal{GMP}_k | k \in A\}$ に変形する．ここで，$R_k (\subset J)$ は k 及び k 以外の他のエージェントが k に割り当てる可能性があるすべてのジョブの集合，$S_j (\subset A)$ はジョブ j が割り当てられる可能性がある全てのエージェント集合である．しかし $\mathcal{GMP}_k$ の一番目の制約条件 (割当制約) には，k 以外のエージェントによって決定される変数 $x_{ij} (i \neq k)$ が含まれているため，割り当ての部分集合を求めることが困難である．そこで，割当制約を緩和するためにラグランジュ緩和を用いる．

2.3. ラグランジュ緩和

ラグランジュ緩和とは緩和した制約にラグランジュ乗数を掛けて目的変数に組み込む事で，単に制約を除去するよりもよい解を得る手法である．適切なラグランジュ乗数を決定するために，微分不可能関数における非線形最適化手法である劣勾配法が用いられるが，この時使用する劣勾配については 3.1.2 に示す．

個々のエージェントに対応する部分問題 $\mathcal{GMP}$ に対し，割当制約を緩和したラグランジュ緩和問題 $\mathcal{LGMP}(\mu)$ は次のように変形される．

$$\mathcal{LGMP}_k \text{ (decide } x_{kj},\ \forall j \in R_k) :$$

$$\text{max.} \sum_{j \in R_k} p_{kj} x_{kj} + \sum_{j \in R_k} \mu_j (\frac{1}{|S_j|} - x_{kj}) \tag{9}$$

$$\text{s.t.} \sum_{j \in R_k} \omega_{kj} x_{kj} \leq c_k, \tag{10}$$

$$x_{kj} \in \{0,1\},\ \forall j \in R_k. \tag{11}$$

ここで μ_j は，ジョブ j に対する実数値パラメータでジョブ j のラグランジュ乗数，即ちジョブ j の価格である．また，$\mu = (\mu_1, \ldots \mu_n)$ はラグランジュ乗数ベクトルと呼ばれる．この時，一般性を失う事無く $S_j \neq \phi$ と仮定できる．ここで決定変数 x_{kj} の値はエージェント k が決定する．すなわち，各ジョブについてそれを選択するか否かの決定権がエージェント k にあるものと仮定すると，この部分問題 $\mathcal{LGMP}_k(\mu)$ は，エージェント k の決定変数 x_{kj} しか含まないため，エージェント k のみにより単独で解決する事ができる．

系全体の問題 $\mathcal{GAP}$ と部分問題 $\mathcal{LGMP}_k(\mu)$ の関係について次の 2 つの命題が成り立つ [2]．これらの命題は GMAP を解くアルゴリズムの設計の上で重要な役割を果たす．

〈 命題 1〉 μ の任意の値に対するすべてのエージェントの $\mathcal{LGMP}_k(\mu)$ の最適値の和は $\mathcal{GAP}$ の最適値の上界値を与える．

〈 命題 2〉 μ のある値において，全てのエージェントが $\mathcal{LGMP}_k(\mu)$ の最適解を得て，且つそれらが割当制約 (2) を全て満たしているとき，それらの最適解は $\mathcal{GAP}$ の最適解を構成している．

3. 分散ラグランジュ緩和プロトコル

GMAP を解く手法として分散ラグランジュ緩和プロトコル (DisLRP) [2] が提案された．これは全てのエージェントが互いの暫定的な解を交換しながら並行してラグランジュ緩和問題 $\mathcal{LGMP}_k(\mu)$ とラグランジュ双対問題を交互に解くアルゴリズムである．

3.1.DisLRP の動作

DisLRP では主にエージェントが自分に関連するジョブから，容量制約を満たしつつ最適な値になるような部分集合を求めるラグランジュ緩和問題と，各エージェントから通知されたジョブの決定変数から，ラグランジュ乗数を更新するラグランジュ双対問題の二つの問題を交互に解く．以降では文献 [2] に従い，ラグランジュ緩和問題を主問題，ラグランジュ双対問題を双対問題と呼ぶ．また，主問題と双対問題を交互に解く 1 回の反復をラウンドと呼び，その反復回数 t をラウンド数と呼ぶ．

3.1.1. ラグランジュ主問題

自身の問題に関連するラグランジュ乗数の値が決定されると，エージェント k はそれぞれの部分問題 $\mathcal{LGMP}_k(\mu)$ を解き，決定変数 $x_{kj} (\forall j \in R_k)$ の値を決定する．これは R_k 内のジョブ (各ジョブ j の効用値は $p_{kj} - \mu_{kj}$，資源消費量は ω_{kj}) を取捨選択して，容

量 c_k の「容器」に効用値の和が最大になるように詰めるナップサック問題を解く事に相当する．

3.1.2. ラグランジュ双対問題

近傍及び自身の決定変数の値が決定されると，エージェント k は，R_k 内の各ジョブ j に対応するラグランジュ乗数 μ_j の値を更新する．このとき，劣勾配法を用いて μ_j の値を更新する．具体的には，エージェント k は，R_k 内の各ジョブ j に対して，近傍から受信した最適解での決定変数の値を基に

$$G_j^{(t)} = 1 - \sum_{k \in S_j} x_{kj} \tag{12}$$

で計算される，緩和されたジョブ j の割当制約に対する劣勾配 $G_j^{(t)}$ と，離散的な時間 (反復回数)t とともに一定の割合 $r(0 < r < 1)$ で減少するステップ長 $l^{(t)}(l^{(t+1)} = rl^{(t)})$ 及び，集合 S_j のサイズを用いて，ジョブ j に関するラグランジュ乗数 μ_j を次のように更新する．

$$\mu_j^{(t+1)} \leftarrow \mu_j^{(t)} - l^{(t)} \frac{G_j}{|S_j|} \tag{13}$$

このとき，μ_j の値はジョブ j に固有のものであり，S_j に属するすべてのエージェントで μ_j に共通の値を代入する事が，$\mathcal{GAP}$ の最適解を構成する必要条件である．これを実現するため，従来手法ではすべてのエージェントのラグランジュ乗数の初期値，ステップ長の初期値 $l^{(0)}$ とその減少率 r にそれぞれ共通の値を設定し，また任意のエージェントは近傍及び自分の t ラウンド目の反復後の決定変数の値を見ながら，次の $(t+1)$ ラウンド目のためのラグランジュ乗数の値を劣勾配法により決定する．これにより，S_j に属するエージェント間で明示的に通信を行う事無く，共通の初期値を共通の規則に従ってラグランジュ乗数を決定し，一貫して μ_j に共通の値を代入する事ができる．

3.1.3. 終了判定

すべてのエージェントが主問題を解き，各近傍の最適解を構成する決定変数の集合を通知した後，まず各ジョブについて割当制約が満たされているかどうかを判定する．その結果，全てのエージェントにおいて，すべてのジョブの割当制約が満たされていれば，命題 2 より系全体の問題 $\mathcal{GAP}$ の最適解 (または制約を満たす下界値となる実行可能解) が得られているため終了する．そうでなければ割当制約が満たされていない各ジョブ j について，S_j 内の全エージェントが劣勾配法によってラグランジュ乗数 $\mu_j^{(t)}$ を $\mu_j^{(t+1)}$ へ更新する．その後，$t \leftarrow t+1$ として全てのエージェントが主問題を解く段階へ移行する．

3.2. 解の収束

命題 1 と命題 2 を満たした時の解は $\mathcal{GAP}$ に対する最適解となるが，そのような割当が必ず見付かるという保証は無い．既存手法 [2] では，$\mathcal{GAP}$ の実行可能解を求めるために，任意のラグランジュ乗数 μ_j の値が共通であるという条件を緩め，S_j 内の各エージェントは μ_j に異なる値を代入してもよい事にし，最適解にはならないかもしれないが実行可能解へ直接収束させる方法を用いている．この手法では単純に，μ_j の更新式 (13) における変化分 $l^{(t)} \frac{G_j}{|S_j|}$ をプラスマイナス最大 $\delta(0 \leq \delta \leq 1)$ の割合でエージェント毎にランダムに変動させている．すなわち，更新式 (13) を次のように置き換える．

$$\mu_j^{(t+1)} \leftarrow \mu_j^{(t)} - (1 + N_\delta) l^{(t)} \frac{G_j}{|S_j|} \tag{14}$$

ここで，N_δ は $[-\delta,\delta]$ で一様分布に従う確率変数である．$\delta = 0$ のとき，更新式 (14) は更新式 (13) と同様のものになるが，δ に 0 以外の値を設定すると，各ジョブ j のラグランジュ乗数 μ_j を，S_j 内のエージェントはそれぞれ異なる値に更新する．各エージェントが独自に値を更新する事で，比較的速く実行可能解へ収束する．この手法を用いる DisLRP は，少ないラウンド数で近似解を求める．また，δ の値を大きくすると，得られる実行可能解の質が平均的に低下するが，同時に実行可能解へ収束するための平均コストは小さくなるという傾向がある．

3.3.DisLRP の問題点

従来の DisLRP では双対問題でジョブにおけるラグランジュ乗数を更新する際，その更新幅を決めるステップ長 $l^{(t)}$ は基本的に定数値である．この更新規則は分散環境での実現が容易である．また，その定数値を適切に選べば，DisLRP は最適値にある程度近い上界値に収束する．しかしながら，その適切な定数値は自明ではなく，設計者が問題例に応じて手動で調整する必要がある．問題の適切な定数値に対して極端に小さい値の場合，ラグランジュ乗数を更新し効用値を更新しても主問題にて選択されるジョブが先ラウンドのものと変わらず，ラウンド数が冗長に増加してしまう．逆に適切な定数値に対して極端に大きい値の場合，個々のエージェントの効用値の更新幅が大きく広がる事によって比較的速い収束が可能だが，本来効用値の低いジョブの効用値が更新する毎に上昇し，実行可能解へ収束するもののその解の精度は低下する．

3.4.ADisLRP

DisLRP に対し，系全体の大域情報を収集してステップ長を適応的に更新する手法である ADisLRP が提案されている [5, 6]．ADisLRP は，生成木を用いてエージェントの決定変数に対応するジョブの効用値を他エージェントに伝搬し，系全体の大域情報を収集する．そして収集した大域情報をもとに，あるラウンド t における各エージェントの主問題の最適解の和を上界値とする．また，複数のエージェントに選択されたジョブは最も効用値の高い選択をしたエージェントに割り当てられるように推定し，その時のジョブの組み合わせの解を下界値とする．このようにして推定した上界値 $ub^{(s)}$ と，下界値 lb をもとに式 (15) によりステップ長を適応的に更新する手法である [8]．

$$l^{(t)} = \pi^{(t)} \cdot \frac{\min_{S \in \{1,\ldots,t\}} ub^{(s)} - lb}{\sum_{j \in J} \{G_j^{(t)}\}^2} \tag{15}$$

ここで $\pi^{(t)}$ は，初期値が 2 で時間と共に幾何的に減少するスカラー量，$g^{(t)}$ はラウンド t における劣勾配である．しかし，この手法では各エージェントが持つジョブの効用値を他のエージェントに通知する必要がある．そのために追加的な通信のコストがかかる．また，エージェントに関連するジョブの効用値を他のエージェントに通知することは，分散最適化において各エージェントのプライバシーを保護する事を研究の動機とする観点において，一歩後退していると考えられる [7].

4. 提案手法

本研究では，DisLRP を改良するための手法を二つ提案する．1 つめの手法では効用値を更新するためのステップ長をエージェントの局所情報を用いて計算し，2 つめの手法ではステップ長を系内の DisLRP の収束状況に応じて動的に変更する．これら二つの手法はいずれも追加的な通信を利用しない．すなわち，エージェントが持ちうる情報のみを利用し，他エージェントに自分のプライバシー情報を伝搬せずに価格更新を行う，という方針に基づく手法である．

4.1. 局所情報によるステップ長の設定

4.1.1. アイディア

従来の DisLRP では効用値更新を行うステップ長に定数値を用いる．適切な定数値をステップ長に設定できれば，DisLRP はよりよい解の精度とラウンド数を得る事ができる．しかし問題毎に適切な定数値は異なるため，実際には，設計者が問題に応じて調整しなければならない．ステップ長が大きすぎる場合は実行可能解の精度は低下し，逆に小さすぎる場合収束に至るラウンド数が増加する．そこで，問題に応じて適切なステップ長を決定するために，エージェントの局所情報を用いる手法を提案する．提案手法では DisLRP におけるジョブの効用値や資源消費量のような，各エージェントが持つ情報からステップ長に代入する数値を算出し，他エージェントから追加的な通信を使って伝搬される情報を用いない．DisLRP では，エージェント k が持つ情報には以下のものがある．

1. エージェント k に関連するジョブ j の効用値 p_{kj}，資源消費量 ω_{kj}，およびエージェント k の資源容量 c_k
2. エージェント k が主問題を解き，求めた決定変数 x_{kj}
3. 関連するエージェント $i(i \neq k)$ から通知された自身に関連するジョブ j の，他のエージェントの決定変数 x_{ij}
4. 双対問題で更新されたラグランジュ乗数 μ_{kj}

この内 1. は自分のみが保有する情報であり，追加的な通信を使わない限り他エージェントに伝搬される事が無い．2. 3. については，主問題から終了判定に移行する際に，通信を用いて近傍のエージェントに通知する情報であるため，他のエージェントにも伝搬し通知され，また他のエージェントから受け取ることにより，関連するエージェント間で共有される．これらは終了判定を行うために必要な情報であり，従来手法 [2] でも用いられる．このため，ジョブに関連する各エージェントの決定変数をステップ長を計算するためのパラメータとして用いる事はエージェントの内部的な情報の漏洩を増加しない．

4. のラグランジュ乗数 μ_{kj} は 1. と同様，自身が更新し利用する情報であるため ADisLRP のように追加的な通信を行わない限り他エージェントに伝搬されないが，全てのエージェントが同じ更新規則を用いて更新を行う場合は全てのエージェント共通の値となっているため，他のエージェントにも知られている情報と言える．しかし，3.2 のように確率変数を用いるなどして，ラグランジュ乗数が個々のエージェントで独自に更新される場合は，個々のエージェントのみが持つ局所情報であると言える．

ADisLRP では大域情報の送受信の際に更新したジョブの価格を通知する．これは各エージェントが劣勾配法によって更新したジョブの価格の変動幅を通知する事に相当する．すなわち，個々のエージェントがどのように価格を更新したかという情報を，他のエージェントは知る．これに対し，先述のように局所情報のみを用いる事で，このような内部的な評価値の漏洩を抑制できると考えられる．これは交渉等において，自身の希望や提案は示すが，内部的な方針を明示的に示さない事に相当すると言える．

4.1.2. 具体的な方法

本研究では，ジョブの効用値と適切なステップ長との間に相関があると予想し，ステップ長 l を導出するための手段として，各エージェント k に関連するジョブ j の効用値 p_{kj} の平均値 $\bar{p_k}$ を用いる．平均値は導出のコストが低く，比較的容易に利用できることから，基本的な手法として用いた．具体的には，DisLRP における主問題・双対問題を解く前に，エージェント k は自身に関連するジョブ j の効用値の和を計算し，それを関連するジョブの数 $|R_k|$ で正規化する．こうして得られたエージェント k に関連するジョブ j の効用値の平均 $\bar{p_k}$ をステップ長として利用する．すなわち，式 (16) のように，ステップ長 l に $\bar{p_k}$ を直接代入する．

$$l_k = \sum_{j \in R_k} p_{kj} \cdot \frac{1}{|R_k|} \tag{16}$$

効用値の平均値は問題中のエージェントに関連するジョブの効用値によって異なるため，各々のステップ長も変動し，更新のためのパラメータは各エージェントで異なる．

この事から，提案手法は更新に用いるパラメータを，各エージェントが個別に異なる値に調整する事によって，部分問題を解く近似解法であると言える．

4.1.3. 期待される効果

平均値を利用したステップ長を用いる事で，問題毎に適応したステップ長を設定し，比較的高い解精度と低いラウンド数を得る事が期待される．また，大域情

報を収集してステップ長を導出する手法 [5] と比較して，提案手法は大域情報を収集しないため，エージェント間の追加的な通信を用いない．そのため，通信にかかるコストを削減でき，エージェントの評価値の漏洩を抑制できる．

4.2. 制約違反数に基づくステップ長の動的変更

4.2.1. アイディア

従来手法ではステップ長をラウンド数に応じて減少させる減少率を用いる．ラグランジュ緩和法によってラグランジュ乗数を更新させる際，減少率を設定し，計算回数につれてラグランジュ乗数 μ の更新幅を縮めてゆく手法が用いらいる [8]．しかし，計算回数が増えても収束が見込めない場合ステップ長 $l^{(t)}$ は下がる一方で，価格の更新幅が小さくなる事により収束性がかえって悪化する可能性がある．本研究では，エージェントに関連するジョブの決定変数 x を利用してステップ長を動的に調整する手法を提案する．この手法では，DisLRP の終了判定において，近傍のエージェント k から通信によって通知されるジョブ j の決定変数 x_{kj} を基に，割当制約 ($\sum_{k \in S_j} x_{kj} = 1, \forall j \in R_k$) に違反しているジョブの数を利用する．すなわちあるラウンドの主問題において，複数のジョブに選択されているか，どのエージェントにも選ばれていないジョブの数を参照する．他のエージェントから通知される決定変数は，DisLRP の終了判定とそれに続く双対問題を解くために必要な情報であり，従来手法でも通信によって伝搬される．そのため，この情報を利用しても従来手法 [2] と比較してエージェントの内部的な情報の漏洩が増加することにはつながらない．

4.2.2. 具体的な方法

割当制約の制約違反数に基づくステップ長の動的な変更では，DisLRP の終了判定と双対問題の動作を変形する．具体的な手順は次の通りである．

1. 終了判定の際，各エージェント k は自身に関連するジョブ j について，自身を含む全てのエージェントからの決定変数 x_{kj} を判定し，割当制約に違反する (複数のエージェントに選択されているか，どのエージェントにも選択されていない) ジョブの数 $E^{(t)}$ をカウントする．但し初期ラウンドのみ ($t = 1$)，最大制約違反数 E_{max} に初期ラウンドの制約違反数 $E^{(1)}$ を代入する．

2. 次の式を用いてステップ長 l を変動させる．

$$l'^{(t)} = l^{(t)} \cdot \frac{E^{(t)}}{E_{max}} \tag{17}$$

3. 双対問題終了後，現在のラウンドの制約違反数 $E^{(t)}$ が最大制約違反数 E_{max} を超えていた場合，最大制約違反数 E_{max} を更新し，主問題へ戻る．

このようにして，ステップ長を制約違反数 $E^{(t)}$ に応じて動的に変更する．直感的には，制約違反数が多いラウンドでは，制約違反数を減らすためにステップ長を大きく設定する事で効用値を大幅に更新し，制約違反数の少ないラウンドでは，効用値を大きく変化させないためにステップ長を小さく設定し効用値を小刻みに更新する．

また，制約違反数 $E^{(t)}$ がそれまでの最大制約違反数 E_{max} を超えた場合，ステップ長を本来の値以上に大きくし，最大制約違反数 E_{max} を更新する．これにより，制約違反数が大きく増加した場合 (局所解を脱出した可能性がある場合)，効用値を大きく更新する事で問題の仕切り直しを行う．次回以降のラウンドでは式 (17) の分母となる最大制約違反数 E_{max} が増加することにより，さらに慎重に効用値を更新する．

4.2.3. 期待される効果

この手法では，初期ラウンドの制約違反数を後々のラウンドの制約違反数が上回り，最大制約違反数が更新される事は稀である事が予備実験により示されており，ステップ長が初期ラウンドの数値を超えて大胆に価格を更新する例は少ない．従って，この手法を用いて価格を更新する際，定数をステップ長として利用する手法と比べて小さいステップ長で価格を更新する事が多いと考えられる．DisLRP にはステップ長の増加に伴い，解の精度が減少する代わりに収束に必要なラウンド数が削減される傾向があり，動的な変更手法を使う事でステップ長は減少し，定数を使用する場合に比べて解の精度と収束するラウンド数が増加する事が予想される．

また，この手法は割当制約違反数 E のみに基づいてステップ長を動的に変更しているが，式 (17) を

$$l'^{(t)} = y \cdot l^{(t)} \cdot \frac{E^{(t)}}{E_{max}} \tag{18}$$

の様に適当なパラメータ y を用いる式に変形する事で，パラメータの調整をある程度行える可能性があると考えられる．

5. 評価

提案手法を採用した DisLRP について，計算機シミュレーションによる実験を行い，その効果を評価した．実験では，GAP ベンチマークの問題例に習い作成した問題を使用した．問題におけるエージェント数，ジョブ数，ジョブの効用値・資源消費量の設定範囲，エージェントの資源容量を指定し，ジョブの効用値と資源消費量を設定範囲以内の自然数からランダムに設定した．本実験ではエージェント数 n を 5, 10，ジョブ数を $5n$，ジョブの効用値・資源消費量の設定範囲 $[1, p]$ を [1, 10], [1, 50]，エージェントの資源容量を $2p$ とした，四通りの GAP をそれぞれ 3 つ作成し，これらを GMAP として DisLRP で解く実験を行った．実験では確率変数 N_δ によるステップ長変動幅のランダムな設定を行うが，$\delta = 1.0$ とし，N_δ は $[-\delta, \delta]$ 上で一様分布に従うものとした．実験におけるラウンド数の上限を 10000 と設定し，実行中に上限を超えた場合，実行を打ち切り，解なしと判定した．また，ステップ長の減少率 r は特に断りが無い限り $r = 1.0$ とし，実行中にステップ長は減少率に

よって減少しないものとした．実験中，エージェント間は 1 対 1 通信を行うと仮定してシミュレーションを行うが，通信における遅延は考慮せず，また通信によりメッセージは欠損しないものとした．

問題中の全てのジョブは一度エージェントに適当に割り当てられたと仮定し，それらが最適解になるようにエージェント自ら割り当て直すという GMAP を作成して実行した．なお，各問題は予め集中型解法により GMAP の最適解を確認した．この GMAP では，各エージェントに最初に割り当てられたジョブは自分を含む系内のエージェントに任意に割り当てる事ができる．即ち，この場合全てのジョブ j について $S_j = A$ である．したがって，全てのエージェント k について $R_k = J$，すなわち，エージェント k の部分問題 $\mathcal{LGMP}_k(\mu)$ には系全体のすべてのジョブが含まれる事になる．このような「完全」な割り当て/被割り当て関係を持つクラスの GMAP は，他のクラスの問題よりも得られる実行可能解の質がやや低いという傾向がある [2]．

5.1. 性能の比較

ジョブの効用値の平均をステップ長として用いる提案手法 1(LDisLRP1)，ステップ長を制約違反数に応じて動的に変更する提案手法 2(LDisLRP2)，大域情報を収集してステップ長を求める ADisLRP を比較した．GMAP をそれぞれ 100 回ずつ解き，得られた解の平均値，収束に至ったラウンド数，収束した回数を集計した結果をそれぞれ図 1〜3 に示す．図 1〜3 の横軸は問題のインスタンス，縦軸はそれぞれ解精度，ラウンド数，収束回数である．グラフ横軸の問題のインスタンスは「(エージェント数)-(ジョブの効用値の範囲)-(問題番号)」により表記する．なお，図 1 と 2 の縦軸は ADisLRP の値を 1 として正規されている．

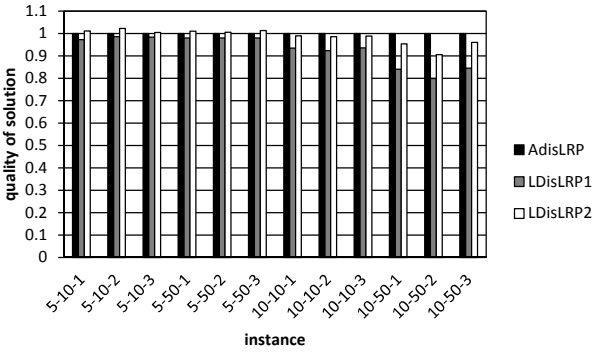

図 1: 解精度の比較

図 1 に示されるように，エージェント数 5 の問題では LDisLRP1 は ADisLRP よりも解の精度が若干小さく，LDisLRP2 は ADisLRP よりも解の精度が上回る結果となった．また，ジョブの効用値の設定範囲を広げた事による影響は顕著ではなく，規模の小さい系において両手法とも効用値のスケールによる影響を受けずに解精度を維持する事が分かった．しかし，エージェント数 10 の問題では LDisLRP1 と LDisLRP2 は，ADisLRP よりも解の精度が下回る結果となった．提案手法では，効用値の範囲を大きくした問題の方が解精度が低下した．エージェント数の増加，ジョブ効用値の範囲の増加が提案手法の解精度を低くする要因であると考えられる．LDisLRP1 と LDisLRP2 を比較すると，どの問題例においても LDisLRP2 の方が解の精度が高く，定数を用いるよりもステップ長を動的に調整する方が解の精度が高い傾向が見られた．

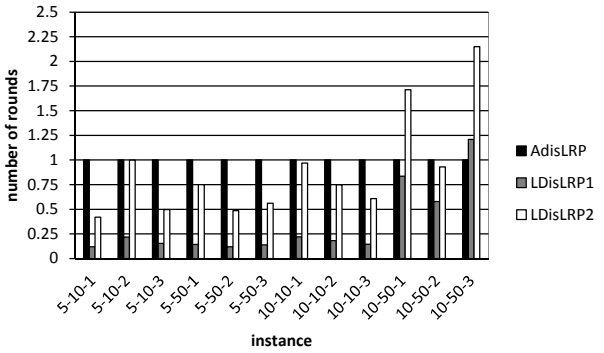

図 2: ラウンド数の比較

図 2 のラウンド数の結果では，エージェント数が少ない，または効用値の範囲が狭い問題例において，LDisLRP1 と LDisLRP2 は ADisLRP を下回る結果となった．特に LDisLRP1 の収束ラウンド数は ADisLRP と比較して多くの場合に 4 分の 1 以下に削減され，平均値をステップ長に利用する事で計算回数が大きく削減された．LDisLRP2 のラウンド数は，一部の結果では ADisLRP とそれほど差がないが，その他の結果では ADisLRP の収束ラウンド数を大きく下回った．その一方で，エージェント数 10，効用値の範囲 [1，50] の問題 10b では，ADisLRP の収束ラウンド数に近い値か，上回る結果となった．この場合の LDisLRP1 の収束ラウンド数は ADisLRP と比較して，ほぼ同等な結果と考えられるが，LDisLRP2 のラウンド数は ADisLRP を大きく上回った．解の精度と同様に，エージェント数の増加，ジョブ効用値のスケールの増加に伴い，ラウンド数も上昇すると考えられる．

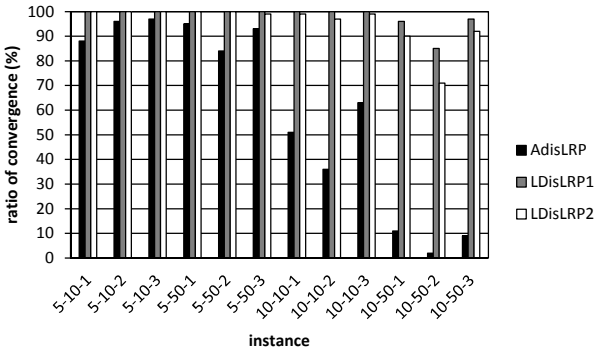

図 3: 収束回数の比較

図 3 の 100 回の試行中に収束した回数では，ADisLRP はエージェント数の増加，効用値スケールの増加に伴い，収束回数が低下した．特に 10b の問題では収束率は 10 分の 1 にまで低下した．LDisLRP1 と LDisLRP2 も同様に収束率は共に減少していく傾向があるが，その減少の度合いは ADisLRP ほどではなく，両者とも収束率が七割を下回らなかった．提案手法 2 つを比較すると，LDisLRP1 の方が上回っており，ラウンド数の比較と同様に，平均値をそのまま用いる方が収束性がよい結果となった．

いずれの解法においても，エージェント数が増加する，またはジョブの設定数値の範囲が増大する事で，収束へのラウンド数が増加し，収束回数が減少した．エージェント数の少ない場合において，数値範囲の増大による性能低下の影響は比較的小さいが，エージェント数の増加によるラウンド数の増加，収束率の減少は無視できないものであると言える．効用値のスケールの増大，エージェント数の増加により提案手法の解の精度は低くなる事が予想される．

5.2. ステップ長の比較

提案手法は，個々のエージェントが持つ局所情報の中からジョブ効用の平均値を用いるという発見的な手法であるため，その有用性の原因を探るために実験的な解析を行った．提案手法 2 つと ADisLRP の計 3 手法において，価格 (ラグランジュ乗数) を決定する数値であるステップ長の統計をとる実験を行った．この実験では，5.1 の実験で使用した GMAP のうち一部を使用した．各問題において 3 手法とも適当な乱数を用いて DisLRP を実行した．得られたステップ長の集計結果を表 1〜3 に示す．それぞれ表 1 が ADisLRP，表 2 が LDisLRP1，表 3 が LDisLRP2 におけるステップ長の集計結果である．表 1， 3 は左から問題 (Instance)，最大値 (Max)，最小値 (Min)，平均値 (Average)，分散 (Variance) である．

Instance	Max	Min	Average	Variance
5-10-3	7.844	0.123	2.798	4.169
5-50-1	41.749	6.441	19.930	86.303
5-50-2	32.656	0.000	4.512	40.405
10-10-1	8.997	0.040	2.026	3.112
10-10-3	9.302	0.512	2.165	2.695
10-50-3	28.149	0.265	7.298	39.569

表 1: ADisLRP におけるステップ長

問題として使用した GMAP は，エージェント数，効用値の設定範囲によって分けられる 4 通りの問題から，少なくとも一つを使用している．また，表 1 の問題 5-50-2 における ADisLRP では，最小値が 0.000 となっているが，これは小数点以下 3 桁で表記できる数値よりも小さい値である事を表す．

表 2 は LDisLRP1 のステップ長となる平均値を表す．LDisLRP1 のステップ長はジョブ効用の平均値を使うので，ラウンド中その値が変わる事は無く，最大値，最小値，平均値は常に同じ値となり，分散は 0 となる．ステップ長の統計としては ADisLRP と大きく異なるが，LDisLRP1 は単に平均値をそのまま使うためである．

Instance	Average	Instance	Average
5-10-3	5.04	5-50-1	26.42
10-10-1	5.534	5-50-2	23.40
10-10-3	5.674	10-50-3	24.724

表 2: LDisLRP1 におけるステップ長

Instance	Max	Min	Average	Variance
5-10-3	5.720	0.337	1.012	1.014
5-50-1	28.160	1.577	4.589	20.097
5-50-2	27.947	1.240	8.758	18.270
10-10-1	6.481	0.136	1.816	1.666
10-10-3	6.427	0.142	1.341	1.595
10-50-3	28.800	0.505	5.335	29.492

表 3: LDisLRP2 におけるステップ長

その一方で LDisLRP2 は，初期値に平均値を代入するため初期ラウンドのステップ長は LDisLRP1 と同様の値になるが，その後は制約違反数に応じて変更されるため，収束状況に応じて変化していく．LDisLRP2 のステップ長は過去のラウンドの最大制約違反数と現在のラウンドの制約違反数によって式 (17) のように決められる．また，多くの場合において，最大制約違反数数は初期ラウンドの制約違反数となり，以降のラウンドで最大制約違反数数が更新される事が稀であるため，ADisLRP と比較するとステップ長はある程度は同様の値をとる．

ADisLRP と LDisLRP2 を比較すると，ほとんどの問題において ADisLRP の方が最大値，平均，分散が高く，ADisLRP の方が大きく広範囲な数値幅のステップ長で価格を更新している事が分かった．LDisLRP2 のステップ長は ADisLRP と比較して狭い範囲でステップ長を更新しているため，ステップ長が急激に変化する事で収束ラウンド数や解精度に影響を及ぼす可能性が低い事が考えられる．先述の通り，LDisLRP2 のステップ長はある程度固定のものとなっている．この特性が 5.1 のような結果の原因の一つである可能性が考えられる．

6. まとめ

本研究では，一般化相互割当問題の解法である分散ラグランジュ緩和プロトコルにおいて，エージェントの局所情報のみを利用した価格更新を用いる 2 つの手法を提案した．また，提案手法により局所的な情報のみを用いても，大域的な情報を利用する ADisLRP に対しほぼ同等な効果が得られることが実験により示された．

ジョブの効用の平均値をステップ長として利用する

手法は，従来手法に比べ解の精度はある程度低下するが，比較的少ない計算回数で収束する．精度の低下は顕著ではなく，許容される場合があると考えられる．系全体の制約違反数に応じてステップ長を動的に変更する手法では，従来手法とほぼ同等の計算回数で，ある程度近い解の精度を得た．

本研究ではジョブの効用値と適切なステップ長に相関があると予想し，ジョブ効用の平均値をステップ長として使う簡単な手法を提案した．その一方で，エージェント数の増加やジョブ効用値のスケールの増加に伴い性能が落ちる可能性も考えられ，単に平均値を適用するのみではなく，より高度な指標を検討する余地がある．今後の課題として，提案手法の理論的な解析および，他の適切な局所情報によりステップ長を設定する手法との比較が挙げられる．平均値以外の統計量，ジョブの資源消費量，過去ラウンドのラグランジュ乗数などのパラメータを活用できる可能性も考えられる．これらについての検討も課題として挙げられる．

謝辞

本研究の一部は，科研費 若手 (B)22700144 および平成 23 年度人工知能研究振興財団研究助成による．

参考文献

[1] 松谷俊宏, 横尾真, 岩崎敦. 架空名義入札に頑健な組み合わせオークションプロトコルの提案と評価: バンドルサイズ優先プロトコル. 情報処理学会論文誌, Vol. 47, No. 5, pp. 1406–1414, 2006.

[2] 平山勝敏. 一般化相互割当問題のための分散ラグランジュ緩和プロトコル. 電子情報通信学会論文誌. D-I, Vol. 88, No. 9, pp. 1269–1277, 2005-09-01.

[3] 平山勝敏. 一般化相互割当問題の上界値を求める分散ラグランジュ緩和プロトコル. 情報処理学会論文誌, Vol. 47, No. 5, pp. 1415–1423, 2006-05-15.

[4] Katsutoshi Hirayama. An α-approximation protocol for the generalized mutual assignment problem. In *Proceeding of the 22nd AAAI Conference on Artificial Intelligence(AAAI-2007)*, pp. 744–749, 2007.

[5] 平山勝敏, 松井俊浩, 横尾真. 分散ラグランジュ緩和プロトコルにおける適応的な価格更新. 人工知能学会論文誌, Vol. 26, No. 1, pp. 59–67, 2011.

[6] 花田研太, 平山勝敏. 過制約な一般化相互割当問題に対する分散ラグランジュ緩和プロトコル. 合同エージェントワークショップ シンポジウム (JAWS-2010) 講演論文集, 10 月 2010.

[7] R Greenstadt, J.P Pearce, and M Tambe. Analysis of privacy loss in distributed constraint optimization. In *Proceedings of the 21th National Conference on Artificial Intelligence(AAAI-2006)*, pp. 647–653, 2006.

[8] M Yagiura and T. Ibaraki. *Generalized assignment problem, in Gonzalez, T.F. ed.Handbook of Approximation Algorithms and Metaeuristics*. Computer & Information Science Series. Hall/CRC, 2006.

[9] 黒田陽之, 平山勝敏. Multi-maxsat: ラグランジュ分解・調整法を用いた weighted max-sat の解法. 電子情報通信学会論文誌 D, Vol. J92-D, No. 1, pp. 51–60, 1 月 2009.