効率的なタスク割り当てのための希望順位戦略の自律的学習法の提案

飯嶋 直輝　杉山 歩未　早野 真史　菅原 俊治

近年の情報技術の発達によって多くの情報が利用可能となり，それによって複数の機能と多様な情報を組み合わせたサービスの提供が可能となっている．これらのサービスは様々な分散環境やその連携で実現できると考えられ，各計算機 (エージェント) にどのタスクを割り当てるかを適切に決定する必要があるが，これにはサービスの要求条件やエージェントの特性を考慮する必要があり，適切かつ効率的なタスク割り当ては困難な課題となる．そこで，本研究では継続的に様々なタスクが発生する環境を想定し，全体の社会的効用に加え，これとは必ずしも整合しないかもしれない選好を個別に考慮できるタスク割り当て問題に対し，各エージェントがシステム全体として求める効用を追求するとともに，それぞれの観点から個々の特徴や実績に基づき，全体に最大限貢献できるタスクの順位づけ戦略を学習する手法を提案し，システム全体として高い効率を実現できることを実験的に示す

1　はじめに

近年の情報技術の発達により，多くの情報が利用可能となり，それによって複数の機能と多様な情報を組み合わせたサービスの提供が可能となっている．これらサービスの実現には様々な分散環境やその連携で実現できると考えられ，各計算機にどのタスクを割り当てるかを適切に決定する必要がある．しかし，これにはサービスの要求条件やエージェントの特性を考慮する必要があり，適切かつ効率的な割り当ては困難な課題であり，多くの研究が行われている [1] [2] [3].

そこで，本研究ではシステム全体の効用を追求するとともに，計算機（エージェント）それぞれの観点から全体（社会的）に最大限貢献できるタスク選択やタスクの順位づけ戦略を学習する方法を提案する．

2　タスク割当問題

エージェントの集合を $A = \{1, \ldots, n\}$ とする．エージェント $i \in A$ は，タスクを処理する能力を表すリソース $R^i = \{r_1^i, \ldots, r_h^i\}$ を持つ．ここで h はリソースの種類を表し，r_k^i はリソースの量を表す．この値が大きいほどタスクの処理能力が高いことを表す．また，離散時間の単位として tick を導入する．

タスク t は三つ組 (S^t, w_t, d_t) で表される．$S^t = \{s_1^i, \ldots, s_h^i\}$ は t を処理するのに必要なリソース量を表す．エージェントはタスク t を処理すると報酬 w_t を得る．本論文では w_t を t の必要リソースの和 $||S^t|| = \sum_{k=1}^{h} s_k^t$ とするが，もちろん社会的な価値などの違う値も考えられる．また本研究では，時間経過で報酬が変化するタスクも考える．i が t を割り当てられたとき，i はタスク処理を開始する．処理にかかる時間は t のリソースと i のリソースの比とし，以下の式で求める．

$$PT^i(t) = \max_{1 \le k \le h} \{\lceil s_k^t / r_k^i \rceil\} \qquad (1)$$

d_t は t のデッドラインを表し，その時刻までに処理を終えなければ t は廃棄される．

l tick においてタスク集合 $T_l = \{t_1, \ldots, t_m\}$ が与えられたとき，エージェント i はデッドラインまでに処理できるタスクに対して，タスクとそのタスクの価値の組の順序集合 $B^i = \langle (t_1^i, v_1^i), \ldots (t_{N_i}^i, v_{N_i}^i) \rangle$ $(m \ge N_i)$ を決定し，マネージャーへ送信する．T_l は l tick におけるシステムのタスクプールと呼ぶ．効

Naoki Iijima, Ayumi Sugiyama, Masashi Hayano, Toshiharu Sugawara, 早稲田大学理工学研究科情報理工・情報通信専攻, Dept. of Information and Computer Science, Waseda University.

用価値 v_k^i はタスクの報酬や処理時間など，共通的で比較可能な値である．t_k^i の希望順位は i の選好戦略 $c^i(t_k^i)$ によって決定するものとする．i の希望順位は他のエージェントと必ずしも整合しない．l tick において A_l^{busy} をタスク処理中のエージェントの集合，A_l^{free} をタスクを割り当てられていないエージェントの集合とする．

毎時間，発生したタスクは T_l に追加される．$\forall i \in A_l^{free}$ は c に従って B^i を決定する．そしてタスクを B に基づいて割り当てを行う．本提案手法では，このタスク割り当て問題を SORA/PO [4] とみなす．これはエージェントの集合 $A = \{1, \cdots, n\}$ とタスクの集合 $T = \{t_1, \cdots, t_m\}$，そしてエージェント $i \in A$ のタスクと，それに対応した価値の順序集合 $B^i = \langle (t_1^i, v_1^i), \ldots (t_N^i, v_N^i) \rangle$ を与えられたとき，以下のように定義される．

定義

タスクの集合を T とし，$B = \{B^i | i \in A\}$ (ただし．$B^i = \langle (t_1^i, v_1^i), \ldots (t_N^i, v_N^i) \rangle$) を希望の集合とするとき，SORA/PO は以下の条件 (3) を満たし，価値の総和を最大にする割り当て $L^*(\subset T \times A)$ を求めることと定義する．

$$J_{all} = \sum_{(t,i) \in L^*} V^i_{o_{L^*}(i)}, \tag{2}$$

subject to

$$\text{For } \forall (t,i) \in L^* \text{ and } 0 < \forall k < o_{L^*}(i),$$
$$\text{if } \exists t_k^i \ s.t. \ a_{L*}(t') = j, \text{ then } \ v^j_{o_{L^*(j)}} \geq v_k^i. \tag{3}$$

これはエージェント i が割り当てられたタスクよりも希望順位の高いタスク t' が他のエージェント j に割り当てられるとき，i の t' に対する価値 v_k^i よりも j の t' に対する価値 $v^j_{o_{L^*(j)}}$ が低くはないことを示している．

この問題は組み合わせ問題であり，最適解を求めるのは困難である．そこで，準最適解を求める SRNF/RD アルゴリズム [4] を使って解を導出する．これは希望順位を考慮し，不満を発生させずに準最適解を求めるアルゴリズムである．本研究では，エージェントは一度に 1 つのタスクしか処理できないとする．(しかし．A^{busy} のエージェントにも負荷に応じて B を宣言できるようにすれば．複数のタスクの割り当ても可能となる．) タスクの処理後，i は報酬 w_t を得る．l tick で割り当てられなかったタスクは，デッドラインによって廃棄されない場合，T_{l+1} に持ち越される．そして新しいタスクが T_{l+1} に追加され，上記の処理を繰り返す．

3 提案手法 : 選好戦略の決定学習

エージェントの選好戦略の集合を C とする．これらの選好戦略は，順序集合 B の決定に利用される．C={HRF,EDF,SPTF,CEF} とし，左から順に報酬の大きい（HRF），デッドラインが近い（EDF），自身にとって処理時間の短い（SPTF），1 tick あたりに得られる報酬 (報酬/処理時間) が大きいタスクを優先する戦略（CEF）を表す．本論文では，環境や負荷，自分の能力に応じてエージェントが C から自分に適切な戦略 c を学習する．はじめにエージェント i は戦略それぞれについてその有用度を示す E 値 $E^i(c)$ を持つ．なお，$E^i(c)$ の初期値は 0 とする毎時間，i は $E^i(c)$ が最大となる選好戦略 c を ε-greedy 法によって戦略を選択し，それに従って B^i を決定する．

まず $E^i(c)$ の学習のために，マネージャーは最近の K 回の (K 個のタスク) 割り当てに関する統計情報として，報酬 w_t，処理開始からデッドラインまでの時間 $d_t - l_s(t)$，処理時間 $l_c(t) - l_s(t)$，を S に保存する，ただし t は処理したタスクを，$l_s(t)$ はタスクの処理開始時間，$l_c(t)$ はタスクの処理終了時間を表す．なお，K を超える場合は古い情報を削除する．この値は後に述べる統計量の計算に利用する．

エージェント $i \in A$ がタスク t を割り当てられたとき，次の式で E 値を更新する．

$$E^i(c) \leftarrow (1-\alpha)E^i(c) + \alpha R(t) \tag{4}$$

ここで $\alpha \in [0,1]$ は学習率であり，R は E 値の報酬 (これを E 報酬と呼ぶ) である．そして，E 報酬を以下のように定義する．

$$R^c = \begin{cases} \pm \dfrac{\theta_c^t - \hat{\mu}_c}{\hat{s}_c} & g_L(i) \neq null \\ -1 & g_L(i) = null \end{cases} \tag{5}$$

$\pm$ は条件により選択する．ここで θ_c^t とはタスク t に関する希望順位を決定するための指標の推定値で，EDF であれば処理開始からデッドラインまでの時間 $d_t - l_s(t)$，HRF ならばタスクの報酬の値 w_t，SPTF

は処理時間 $l_c(t) - l_s(t)$，CEF は $w_t/(l_c - l_s)$ の推定値となる．$\hat{\mu}_c$，$\hat{s}_c$ はそれぞれ指標を標準化するための標本平均と不偏標本標準偏差であり，これらはマネージャーが S から以下の式で計算し，タスク t の割り当ての際に t とともにエージェントに送る．

$$\hat{\mu}_c = \frac{1}{K}\sum_{s\in S}\hat{\theta}_c^s$$

$$\hat{s}_c = \frac{1}{K-1}\left(\sum_{s\in S}(\hat{\theta}_c^s - \hat{\mu}_c)^2\right)$$

従って R^c はそれぞれの指標を表す標準化された統計量となる．このように E 報酬を与えることで，エージェントは自分にとって得意な戦略を学習することができる．たとえば EDF 戦略の場合，他のエージェントよりもより $d_t - l_s(t)$ が小さいタスクを処理するほど E 報酬が大きくなる (この場合は式 (5) で $-$ をとる)．また，エージェントはそれぞれ異なる戦略を持つことになる．E 値は Q 値の概念に近いが，報酬は共通ではなく，各戦略ごとに表した統計量を標準化したもので，その戦略による全体の位置付けを表す．

4 評価実験

4.1 実験環境

本実験の目的は，時間経過に伴いタスクの性質が変化する環境において，選好戦略の学習が柔軟にこれに対応でき，高い効率を実現することを示すことである．

本実験ではエージェント数 $|A|$ を 300 とし，エージェントとタスクのリソースの種類数は 3 とする．エージェントは所持する全てのリソースが多いものと少ないものの 2 種類存在し，$1 \le k \le 3$ について

$$5 \le r_k^i \le 7 \text{ or } 2 \le r_k^i \le 4 \qquad (6)$$

の範囲でランダムで設定する．また，タスクは以下の範囲で要求リソースをランダムに設定する．

$$30 \le s_k^i \le 60\ (1 \le k \le 3) \qquad (7)$$

本実験では，SORA/PO における価値はタスクの効用値とし，システムとしてこれを最大化することを目的とする．タスクの報酬値については 2 種類仮定し，第一をデッドラインまでに処理できれば一定の効用値が得られるタスクとし，これをタイプ 1 と呼ぶ．タイプ 2 のタスクはデッドラインに近くに従って報酬値が減少するタスクであり，以下のように定義する．

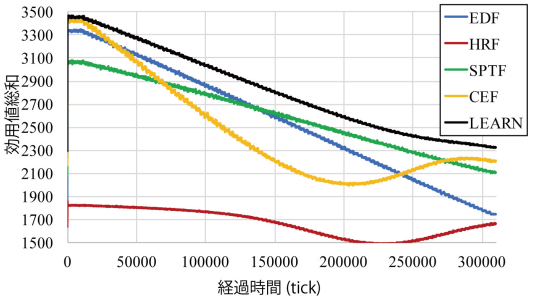

図 1　1 tick あたりの平均効用値の推移

$$w_t = \sum_{k=1}^{h} s_k^t \times \left(1 - \frac{l_c(t) - l_g(t)}{2(d_t - l_g(t))}\right) \qquad (8)$$

ここで $l_g(t)$ はタスクが発生した時間を表す．これは，効用値は時間とともに線形に減少し，デッドラインの直前では半分になる．

タスクは時間経過とともにポアソン分布 $p_0(\lambda)$ に従って発生し，タスクプールに追加される．1 tick に平均で発生するタスク数 λ は，どの戦略でも廃棄タスクが現れ始める値の 26 と設定した．タスクのデッドライン d_t は 15 から 25 の間でランダムな値とする．タスクを処理していないエージェント $i \in A_l^{free}$ は選好戦略に基づき希望リスト $B^i = \langle (t_1^i, v_1^i), \ldots (t_N^i, v_N^i)\rangle$ を宣言する．N_i を大きくすると割り当て対象も増加するが，計算コストも増える．ここでは $N_i = 5$ とした．また，$K = 50$ とし，学習率 $\alpha = 0.1$，$\varepsilon = 0.1$ とする．

本実験では，全体のシミュレーション時間を 310000 tick とし，初めの 10000 tick を初期学習の時間とする．初めは全てのタスクがタイプ 1 のタスクであるが，10000 tick 後は 3000 tick 毎にタイプ 2 の発生する割合を 1%ずつ増加させる．以下に示す実験結果は 50 回の試行の平均値をとったものである．

4.2 効用値とエージェントの戦略の推移

図 1 に 1 tick あたりに獲得したタスクの効用値の総和の推移を，図 2 に提案手法を実装したエージェントが選択した選好戦略の割合の推移を示す．図中で学習する提案手法に関してはラベルを LEARN と

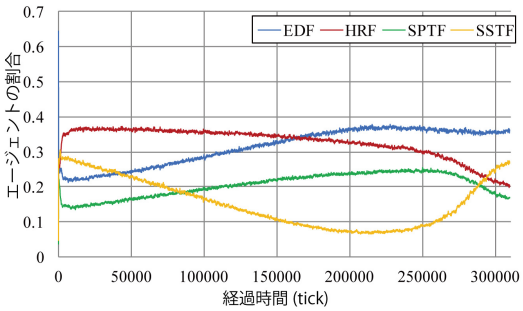

図 2　エージェントの戦略推移

し，以後提案手法を LEARN と呼ぶ．また，ラベル EDF，HRF，SPTF，CRF はそれぞれの選好戦略に統一したときの結果を示す．

まず単一戦略に着目すると，図 1 から環境の変化につれて効用値の総和を最大化する戦略も変化することがわかる．例えば CEF は，初めは一番良く，環境の変化に従って順位が下がるが，最終的には再び一番目に戻る．これは，CEF は効用値を処理時間で割った値を報酬とするので，タイプ 2 のタスク処理で得られる報酬は少なく，タイプ 2 のタスクの増加に伴い，エージェントはタイプ 1 のタスクに集中し，かつタイプ 2 のタスクは廃棄されやすくなり，効用値が低くなる．しかし，タイプ 2 のタスクのみに近づくと希望の偏りがなくなり，再び効用値が上がる．

他方，提案手法に着目すると，図 1 で LEARN はタスクのタイプに関わらず常に単一戦略よりも効用値の総和が高くなった．これは，単一の戦略でなく各エージェントが個々の戦略を貢献度に応じて自律的に選択したためと言える．

図 2 を見ると，戦略を学習するエージェントは環境によって選択する選好戦略の割合が異なることがわかる．例えば，HRF は時間経過に伴い選択される割合が徐々に減少するが，EDF は時間経過に伴い徐々に選ばれる割合が増加し，200000 tick あたりから停滞している．SPTF は初めは選択される割合が上昇するが，250000 tick あたりから減少する．逆に SSTF は初めは選択される割合が減少するが，250000 tick あたりから上昇する．全てのエージェントが同じ戦略にならないのは，同じタスクに希望が集中し，タスクが割り当てられずに負の報酬を得るのを回避するためである．そして，タイプ 2 のタスクが増えるにつれて CEF を選択するエージェントが減少する．これは CEF のエージェントがタイプ 1 のタスクに集中して競合が発生することが原因である．このとき，他の戦略に比べて SPTF のエージェントがより増加している．これは能力の低いエージェントと比べて処理時間が短いため，比較的大きな報酬を得やすいからである．最後にタイプ 2 のタスクのみになると，CEF を選択しても競合が減るため CEF を選択するエージェントの割合が増加する．

5　まとめと今後の課題

本研究ではエージェントがタスクに対して，自身の能力や周囲の環境を考慮して自分のタスクへの希望を宣言できるタスク割り当て問題を提案した．その問題に対し，エージェントが周囲の状況とシステムへの貢献度を考慮して自分の戦略を自律的に選択する手法を提案した．実験によって，戦略を学習することによってエージェントが変化する環境に柔軟に戦略を変更し，全体の効率を上げることを示した．

今後の課題としては環境を変えて実験を行うこと，またエージェントの希望の宣言の際に通信時間を考慮することなどである．

参 考 文 献

[1] Y. Jiang, “A survey of task allocation and load balancing in distributed systems,” IEEE Transactions on Parallel and Distributed Systems, vol.27, no.2, pp.585–599, 2016.

[2] H. Izakian, A. Abraham, and B.T. Ladani, “An auction method for resource allocation in computational grids,” Future Generation Computer Systems, vol.26, no.2, pp.228–235, 2010.

[3] H. Aziz, P. Biró, J. Lang, J. Lesca, and J. Monnot, “Optimal reallocation under additive and ordinal preferences,” Proc. of the 2016 International Conference on Autonomous Agents & Multiagent Systems, pp.402–410, International Foundation for Autonomous Agents and Multiagent Systems, 2016.

[4] K. Saito and T. Sugawara, “Single-object resource allocation in multiple bid declaration with preferential order,” Computer and Information Science (ICIS), 2015 IEEE/ACIS 14th International Conference on, pp.341–347, June 2015.