

自律ロボット・IoT 機器用メッセージ指向ミドルウェア上の非集中型最適化基盤

代表研究者 松井俊浩 名古屋工業大学大学院工学研究科知能情報プログラム 准教授

1 概要

ロボット用ソフトウェアプラットフォーム ROS2 の DDS [1]や IoT 機器の通信に用いられる MQTT [2]などの、通信フレームワークでの活用に特化した、装置間の分散協調アルゴリズムのソフトウェア基盤技術の開拓を目指す。その主要な構成要素として、最適化手法の実装技術と有効性について検討した。特に、分散制約最適化問題の解法を実システム上に実装するうえで必要となる、低 QoS 通信におけるメッセージ損失への対処を含む実時間指向の解法、出版-購読型通信モデル上への適用を検討し、シミュレーション、実装、評価環境を試作し、実験により検証した。

2 背景

2-1 分散協調処理と分散制約最適化

(1) 分散協調処理における最適化問題

自律ロボット/ドローン群による貨物搬送や広域観測、スマート家電/センサ機器を制御する IoT 機器群によるエッジコンピューティングや連合学習など、末端機器の能力向上と情報の非集約の要求から、分散協調処理の高度化が模索されている。系の分散制御に付随する、資源割り当てや行動計画の問題解決のために、分散処理に基づく最適化アルゴリズムの応用が研究されている。人工知能分野の「マルチエージェントシステム」の研究は、人や装置を抽象化した「エージェント」群の相互作用を扱う分野として広く研究されてきた。エージェント群を協調させる基礎技術に、その要求を系に分散した最適化問題として記述しメッセージ通信を伴う分散アルゴリズムにより解決する「分散最適化」手法がある。協調をモデル化した問題と分散最適化アルゴリズムが提案されてきた。本研究では、汎用的な離散最適化問題により協調問題を記述し非集中型の解法により解決する、分散制約最適化問題 [3]とその解法に注目する。

(2) 分散制約最適化問題

分散制約最適化問題 [3]は複数エージェントシステム上の協調問題を、エージェントに分散して配置された制約最適化問題(離散組み合わせ最適化問題)として記述し、非集中型の解法により解決する。

基礎的な分散制約最適化問題は、 $\langle A, X, D, F \rangle$ により定義される。 A はエージェントの集合、 X は変数の集合、 D は変数の地域(領域)の集合、 F は目的関数の集合である。変数 $x_i \in X$ は値域 $D_i \in D$ の値を取る。制約条件は目的関数として記述される。基礎的な問題では、各エージェント $a_i \in A$ は単一の変数 x_i を持ち、目的関数は二項関数である。目的関数 $f_{i,j}(x_i, x_j)$ により、変数 x_i, x_j の値の組み合わせの評価値が定義される。ここでは、系全体の目的関数の合計値を最大化する。エージェントは自変数値と関連する部分問題の知識を持ち、自変数値を決定する。エージェント間のメッセージ通信を伴う分散最適化アルゴリズムにより、最適解を求める。

(3) 解法

分散制約最適化問題の解法は、逐次型の最適化手法を基礎とする分散アルゴリズムである。木探索、動的計画法、局所探索、確率的/サンプリング解法、確率伝播、進化的計算、ラグランジュ双対など、多数の解法が提案され、最適解を得る完全解法と、準最適解や近似解を得る不完全解法に大別される [3]。制約密度が高く大規模な問題では、最適解を求めることは一般には困難であり、不完全解法が現実的である。

代表的な解法の幾つかは、変数間の関係を反映する木構造に基づく木探索や動的計画法などの比較的複雑な処理を伴う。そのため、応用・実装の初期検討では、簡素だがプログラミングが比較的容易な局所探索型の不完全解法を用いる [4] [5]。本研究では、最も基本的な決定論的山登り法 MGM [6]を用いる検討から着手し、エージェント間の木構造による集計を伴う解法として進化的計算 AED [7]についても検討した。以降に、各解法の概要を述べる。

(4) MGM

MGM [6]は、分散制約最適化問題の基礎的な解法であり、最良解を常時保持しつつ段階的に改善してゆく、anytime 性を持つ。この解法では、各エージェントが目的関数で関係する近傍のエージェントと情報交換し、決定論的な山登り法に基づき局所探索を実行する。

各エージェントは、近傍エージェントから受信した変数値からなる部分解を保持するビューを持つ。ビューの内容は、受信したメッセージにより更新される。この部分解と自身の変数値に基づいて、各エージェントは下記の各段階の処理を同期しつつ反復する。

1. 初期状態ではビューは空であるため、自変数値の初期値を任意に決定する。
2. 現在の自変数値を、近傍エージェントに同報する。
3. 近傍エージェントの現在の自変数値を受信し、ビューを更新する。
4. 現在のビューに含まれる近傍エージェントの変数値と自変数からなる部分解の下で、部分解の効用を局所的に最大化する自変数値を探索する。
5. そのときの、効用の増加分を求め、近傍エージェントに同報する。
6. 近傍エージェントから受信した効用の増加分と、自身が効用を最大化した場合の増加分を比較する。
7. 自身が最大化した場合の効用の増加分が、全ての近傍エージェントの増加分よりも大きければ、対応する自変数値を選択する。
8. 上記2の手順から解法を反復する。

MGM は簡素だが扱いが容易であり、anytime 性を持つため、応用を模索する初期段階の検討では、基礎的な解法として用いられる [4] [5]。

(5) AED

AED [7]は進化的計算に基づく解法であり、anytime 性を持つ。この解法は近傍エージェントとの情報交換に基づく局所的な計算と、全エージェントを含む全域木に基づく大域的な計算からなる。解法の各段階でエージェントは同期しつつ処理を進める。

各エージェントは、全エージェントの変数についての、大域的な解候補群である個体群を持ち、解法過程では、自変数と目的関数で関係する近傍の範囲の部分解の知識のみに従い、近傍エージェントと協調しつつ自変数のみを更新する。各個体にはフィットネス値(大域的効用値)が付加される。これらの大域的情報は、初期化処理により与えられる。エージェントは大域解の情報と評価値のからなる個体群を持つが、自身が関連する部分解のみを考慮し自変数を変更することしかできず、また、評価値も局所解についての差分を更新するのみである。

各エージェントは解法の各段階でメッセージ交換する近傍のエージェントとバリア同期しつつ、下記の各段階からなる処理を反復する。

1. 各エージェントは自身の個体群から部分個体群をサンプリングし選択する。次に、選択した個体群に含まれる自変数値をサンプリングして変更し、解の改善を試みる。さらに、部分個体群の改善を、コスト関数で関係する近傍エージェントに送信し、近傍エージェントの変数値についての依頼する。
2. 近傍エージェントは、受信した部分 体群のそれぞれについて、自変数値を最適化し、送信元のエージェントに返送する。
3. 各エージェントは、近傍エージェントによって更新された部分個体群を受信し、元の個体群に加える。

さらに、自身が持つ個体群をサンプリングしてフィルタし、個体群の数を一定に維持する。

上記の近傍エージェントとの情報交換に基づく確率的局所探索だけでは、解を局所的しか改善できない。そこで、解の多様性を維持するために、各エージェントは、定期的の一部の個体群を近傍エージェントに移民させる。

1. 各エージェントは局所的な解の改善のための各段階の処理を一定回数繰り返す度に、部分個体群をサンプリングし、近傍エージェントに送信して移民させる。
2. 移民させる部分個体群を受信したエージェントは、部分個体群を自身の個体群に加える。さらに、自身が持つ個体群をサンプリングしてフィルタし、個体群の数を一定に維持する。

これらの処理において、各エージェントは大域解とその評価値からなる個体群を維持しつつ、局所的に改良し続ける。それらの個体群から、定期的に最良解を選択し、各エージェントの変数値を大域的な最良解に同期することで、anytime 性/実時間性がある解法として活用できる。そのために、エージェント群は、全域木に従う大域的な計算を解探索の背後で同期的に実行し、最適な個体を選択し、自変数値を更新する。

1. 各エージェントは、自身の最良の個体を、全域木に従ってボトムアップに伝搬する。
2. 根ノードのエージェントは大域的に最良の個体を選択し、全域木に従って、トップダウンに解の決定を伝達する。
3. 各エージェントは、全域木における、自身の位置と、同期のタイミングに関する知識により、同一のタイミングで、自変数値を最良解に更新する。

全域木は解法全体で anytime 解を得るために利用できるものであれば良く、木の深さを抑制するために深さ優先探索木が用いられる。

本研究では次の理由により AED を実装方法に関する初期検討の対象にした。1) anytime 性を持つ。2) エージェント/変数間の近傍関係に基づく全域木を集計/同期に用いる解法である。3) 処理の段階が多い点で比較的複雑な解法である。4) 大域的な最適化の機会を持ち、ある程度の制約密度の問題に対応できる確率的解法である。

2-2 出版-購読型通信

(1) 通信モデルと構成要素

出版-購読型の通信モデルは、エッジ機器間、エッジクラウド間の通信基盤として普及し、ロボット用プラットフォーム ROS2 の DDS [1], IoT 機器にも利用される MQTT [2]などに用いられている。基礎的な出版-購読型通信モデルは、出版者と購読者から構成される。出版者はメッセージを同報し、購読者は自身に関連するトピックを持つ特定のメッセージを受信する。また、出版者と購読者の間にブローカが介在し、出版社がブローカに出版したメッセージを、購読者が受信する方式も用いられる。ROS2 の DDS の基本的な構成には、明示的なブローカは無く、LAN 上の同報通信に基づく。MQTT はブローカを介在する方式である。

(2) 分散制約最適化手法の実装と課題

これらの出版-購読型の通信基盤を用いるエッジ機器間の協調に、分散制約最適化手法を活用するためには、通信基盤に応じた解法の実装方法の検討と検証が必要である。また、移動ロボットなど、一次的な通信の遮蔽が起こりうる環境や、実時間性が優先される環境では、メッセージの損失を許容する、低 QoS、ベストエフォートの通信が用いられる。エージェントの一部が一時的に欠落した状況においても、その影響を抑制しつつ、解法を実行することが求められる。

また、ブローカが介在する場合は、地理・ネットワークの考慮と、情報流通と負荷の分散のために、複数のブローカを用いることが考えられ、オーバレイネットワーク上のブローカとエージェント・出版-購読クライアントの配置を最適化することも考えられる。このような課題を考慮した、ROS2 や IoT 機器向け通信で用

いる、メッセージ指向通信上での活用に特化した、分散最適化手法のミドルウェア基盤を模索する機会がある。

3 目的と検討事項

上述の分散制約最適化手法の実装と課題に挙げた構想の基礎検討として、1) 分散制約最適化手法の出版-購読型の通信基盤上での実装、2) 低 QoS 通信環境下でメッセージの損失を許容する実時間型解法、3) 複数ブローカが介在するオーバレイ網上での実装について検討し、その実装方法と有効性を検証することを目的とする。また、4) 実機において、シミュレーションと同等の評価をするための実験環境を開発する。実装環境として、ROS2/DDS, MQTT を対象とする。

4 低 QoS 環境下の実時間的解法

4-1 メッセージ損失を許容する実時間型解法

低 QoS 環境下で、一時的なメッセージ損失により、疎通が途絶えるエージェントを許容して計算を進める解法について検討した。解法により、メッセージ損失の影響と対策が異なる。本研究では、最も基本的な決定論的山登り法 MGM を用いる検討から着手し、木構造による集計を伴う解法として進化的計算 AED についても検討した。

4-2 MGM におけるメッセージ損失への対処

メッセージの欠落が起こりうる環境では、継続的にメッセージを再送することで受信者への伝達の機会を得る。このために、解法の計算の処理の間隔よりも、再送信の処理の間隔を短く設定し、複数回の送信を試みる機会を得る。メッセージが欠落する頻度に対して、十分な回数の発行の再試行が必要となるが、実時間性を得るために、メッセージの発行の再試行は、ある回数までとする。

再送回数の上限までにメッセージが受信できなければ、解法の各段階のバリア同期処理は達成できず、解法を継続できないエージェントが生じる。逆に、メッセージを伝達できなかったエージェントが近傍エージェントからのメッセージをすべて受信し、次の段階に進む場合もありうる。これらの状況により、系全体では解法の進行が整合せず終了しないことが問題となる。

実時間性を確保するために、疎通が途絶えたエージェントを一時的に無視する。近傍エージェントからのメッセージが欠落した場合に、その近傍エージェントを除外して処理を継続することで、系全体での解法の中断を避ける。各エージェントは、ある期間まで待機しても近傍からのメッセージが得られない場合、そのエージェントを協調の対象から除外して処理を継続する。

一部の近傍エージェントからの情報が欠落した場合の対処は、基本的には、そのエージェント以外の情報を集計しつつ、欠落したエージェントの状況を推測することである。

MGM では、各エージェントは自変数値を決定し、近傍エージェント間で現在の変数値を共有し、現在の部分解を評価するため、未受信の近傍エージェントの変数値についての評価の方法に選択肢がある。次の 3 種類の基礎的な方法について検討した。

- 継続：ビューに残っている最新の受信済みの変数値を用いる。
- 上界：未決定の変数について、相手のエージェントが自変数と関係する目的関数値を最大化すると仮定する、楽観的な戦略を用いる。
- 下界：未決定の変数について、相手のエージェントが自変数と関係する目的関数値を最小化すると仮定する、悲観的な戦略を用いる。

表 1 シミュレーションにおける MGM の解品質 [8]

prb.	alg.	lcl. eval.	utility	last update		#ignored. nbr. at last		
				#iter.	#sys. cyc.	min.	ave.	max
MATCHING	BASE	-	795.5	8.1	35.8	-	-	-
	IGNORE	Inherit	614.4	3.6	16.2	0.3	3.0	7.1
		LB	653.7	3.5	18.0	"	"	"
		UB	612.2	3.7	18.9	"	"	"
	RCVCMT	Inherit	783.4	9.0	37.8	0.3	3.0	7.1
		LB	818.4	16.7	70.8	"	"	"
		UB	801.7	22.5	94	"	"	"
	REJOIN	Inherit	783.8	9.1	38	0	1.5	4.6
		LB	817.1	17.0	72	0.05	2.3	5.9
		UB	803.7	22.5	94.0	"	"	"
RANDOM	BASE	-	1072.7	6.1	28.3	-	-	-
	IGNORE	Inherit	1022.8	3.8	17.4	0.3	3.0	7.1
		LB	1020.7	3.8	19.2	"	"	"
		UB	1041.4	4.1	20.4	"	"	"
	RCVCMT	Inherit	1073.2	7.6	32.2	0.3	3.0	7.1
		LB	1066.8	24	100	"	"	"
		UB	1075.4	24.0	99.9	"	"	"
	REJOIN	Inherit	1073.7	8.0	33.8	0	1.5	4.6
		LB	1067.9	24.0	99.9	0.05	2.3	5.9
		UB	1075.9	24.0	99.96	"	"	"

これらのどれが有効か否かについては、問題やエージェントの性質に依存する面がある。

さらに、疎通が途絶えたエージェントを一時的に無視した後、どのように扱うかについて、次の 3 種類の方法についても検討した。

- 無視継続：疎通が途絶えたエージェントを協調に復帰させず、解法の終了まで無視しつづける。上記の未受信の近傍の変数値に関する戦略をとり続ける。
- コミット情報傍聴：協調に復帰させないが、疎通が途絶えたエージェントが、独立して解法を進めた結果として自変数値の決定し、同報するメッセージを受信できたときに、その変数値を利用する。
- 復帰：疎通が途絶えたエージェントからのメッセージを受信でき、解法の処理の段階を整合させる再同期が可能なタイミングがあったときに、協調に復帰させる。

4-3 拡張された MGM の評価

シミュレーション環境において提案手法を評価した [8]。50 エージェント/変数、変数の値域のサイズ 3、目的関数の数を 150 とした。目的関数の 2 項関数の効用値として次の 2 種類を用いた。RANDOM：一様分布乱数に基づく [1, 10] の整数。MATCHING：2 変数値が同一のときの効用値は RANDOM と同様、そうでなければゼロ。

受信、送信、計算の各処理の頻度の比率を等倍とした。25 エージェントが低通信品質を伴うものとし、確率 0.5 で発行メッセージが損なわれるとした。2 回以上連続して疎通が損なわれたエージェントを無視することとした。

疎通が損なわれた後のエージェントの扱いについて、次の戦略を比較した。IGNORE：無視継続、RCVCMT：コミット情報傍受、REJOIN：復帰。また、基準として、BASE：メッセージ損失が無い場合の結果も示す。未

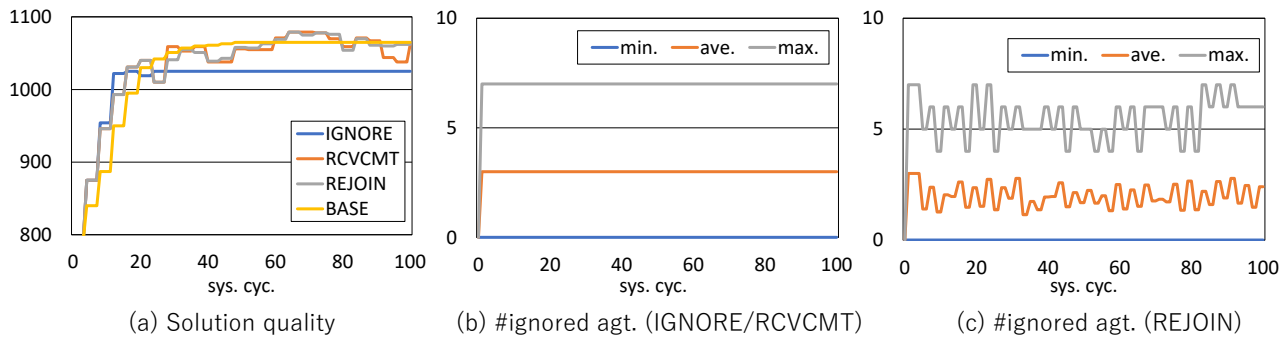


図 1 シミュレーションにおける MGM の推移 (UB) [8]

受信の近傍エージェントの変数値についての評価について、次の戦略を比較した。Inherit: 継続, LB: 下界, UB: 上界。

表 1 に、解品質などの結果を示す。#iter は解法の反復回数であり、#sys. cyc. は各コールバック処理を呼び出すシミュレーション系全体のサイクルの実行回数である。#ignored nbr. at last は、カットオフサイクルにおいて各エージェントに無視されていた近傍エージェント数の平均値である。メッセージ損失がある状況でも各手法は処理を継続した。解品質は基準である BASE よりは低い場合と若干上回る場合があった。これは、決定論的解法に雑音に加わり、確率的探索となった影響と考えられる。

未受信の変数値を推定する戦略については、RANDOM の場合は、楽観的な UB が比較的良好であった一方で、エージェント間の合意の機会が少ない MATCHING では、悲観的な LB が有効な場合が見られた。また、無視されたエージェントの扱いについては、傍受した情報を使う RVCMT と、可能であれば協調に復帰させる REJOIN の解品質の差は僅かであった。近傍エージェントの行動を観測できれば、解品質を改善する機会があると考えられる。

図 1 に、解法の anytime 曲線の例を示す。メッセージ損失による雑音のために正確な anytime 性は損なわれているが、疎通が途絶えたエージェントに対処する手法では、解品質が基準の解法の付近で振動/収束する傾向も見られた。また、再同期の機会が制限される状況もあり、協調の回復をより重視する観点からは改良の余地があると考えられる。

4-4 AED におけるメッセージ損失への対処

AED を上述の MGM の拡張と同様の枠組みにより実装することを想定し、メッセージの再送、疎通が途絶えたエージェントの一時的な無視などの基本的な処理は同様とする。

MGM の場合は、解の情報が欠落することに対して、相手のエージェントが取りうる行動を推測することで、リスクまたは機会の改善を試みる方法をとったが、AED は各エージェントが初期化処理により与えられた完全解の個体群を持ち、それを部分的に改善するため、解の情報の欠落や不整合を起こすことは無い。そのため、解のサンプリング、改良や移民においては、基本的な影響は、探索の遅れや局所解への収束の可能性などの解品質にかかわると考えられる。

その一方で、全域木に基づく、集計と最良解の決定は、その故障が全体的な意思決定に影響するため、最良解を単調に更新し続ける anytime 特性が損なわれることが問題となると考えられる。

そこで、処理の各段階のメッセージ損失について、以下のように対処する。大域的クロックの機構と初期個体群の共有知識を仮定する。

各エージェントがサンプリングし改善を試みた部分個体群のさらなる改善を近傍エージェントに依頼するメッセージが失われた場合は、近傍エージェントによる個体群の改善は行われない。

従って、その応答のメッセージも損なわれる。移民のメッセージが損なわれた場合は、移民が行われない。

最良解の選択と決定のための全域木に沿ったメッセージのボトムアップ、トップダウンな伝搬が損なわれた場合は、そのメッセージにより更新されるはずだった情報は古いままとなる。

ボトムアップに伝搬する最良解の選択のためのメッセージの損失は、次のタイミングでの再送により解消する。

表 2 シミュレーションにおける AED の解品質 [9]

Alg.		Final utility	Nonmonotonicity	
Rejoin	Commit		#case	sum. util.
Base w/o msg. loss		1091.8	0	0
No	Org.	966.4	12.3	-182.1
	Lcl. mono	974.0	1.5	-15.6
	Nbr. best	984.8	1.2	-12.8
Yes	Org.	1092.2	0.13	-1.6
	Lcl. mono	1092.2	0.03	-0.5
	Nbr. best	1092.2	0.03	-0.3

トップダウンに伝搬する最良解の更新のためのメッセージの損失は、次のタイミングでの再送により解消する。しかし、この場合は、各エージェントの情報が他のエージェントのものと整合しないままで自変数値を変更するため、最良解を単調に更新する anytime 特性が失われる場合が生じる。

これらの妥協により、解法の実行は継続できるが、最良解の更新が整合しない場合が生じるため、その緩和策として以下の効果を検証する。

- 局所的な単調性の強制：各エージェントは、自身が把握している最良解のコスト値が単調に減少しない場合は、解を更新しない。
- 近傍エージェントの最良解からの補完と単調性の強制：上記に加えて、各エージェントが近傍のエージェントから最良解を受信し、それらから最良のものを選択する。この通信のタイミングは、全域木に従うメッセージ伝搬と同じバリア同期のタイミングとし、一部のメッセージは、全域木に従うメッセージ伝搬にピギーバックすると仮定する。

4-5 拡張された AED の評価

シミュレーション環境において提案手法を実験により評価した [9]。50 エージェント/変数、変数の値域のサイズ 3、目的関数の数を 150 とした。制約に対応する 2 項関数は一様分布の乱数に基づく [1, 10] の整数の効用値とした。AED の進化的計算のパラメータは従来研究を参考に同一の設定とした。25 エージェントが低通信品質を伴うものとし、確率 0.5 で発行メッセージが損なわれるとした。出版：購読：処理のコールバックの間隔を 1：1：10 とした。購読のタイミングにおいて、2 回以上メッセージが損失したエージェントを、無視することとした。

無視されたエージェントを通信回復後に協調に復帰させるか否か(Rejoin)の各場合を評価した。AED の系全体の整合性において、メッセージ損失の影響を受ける、最良解の同期(Commit)について、次の場合を評価した。Org.：従来手法のままで可能な情報のみ処理する。Lcl. mono.：各エージェントは、適合度値の単調性が失われる解にコミットしない。Nbr. best：各エージェントは、Lcl. mono に先立って、近傍エージェントから受信した、それらがコミットしようとしている最良解のうち、最良のものを選択する。

表 2 に、低 QoS 出版-購読通信系上の AED の実装を模倣したシミュレーション結果を示す。Base w/o msg. loss はメッセージ損失が無い条件における AED を表す。この設定では、メッセージが欠落するエージェント数と、その頻度が比較的大きいため、協調への復帰(Rejoin)を無効にすると、解法の反復が中断する機会が比較的多くなり、解品質に影響した。その一方で、再送の機会が比較的大きいため、協調への復帰を有効にすると、解法の反復は中断することなく進行した。最良解のコミット処理におけるメッセージ損失の影響の緩和策については、Org., Lcl. mono., Nbr. best, の順にさらに解品質が向上し、後者ほど平均的には有効である傾向が見られた。また、最良解が非単調であった回数(¥#case)および非単調な量の合計(sum. util.)の改善の効果についても同様であった。従来検討した基礎的な局所探索手法 MGM の同様の段階の検証の結果と比較すると、解精度の理論的性質としては、AED の方が良好であるが、進化的計算特有の処理とメッセージのオーバーヘッドの抑制、MGM とは異なるメッセージ損失の影響と対処方法などに、実装においては特有の

検討が必要となることが示唆される。

5 ROS2/DDS 環境への実装と検証

5-1 解法の実装モデル

出版-購読型の通信基盤のフレームワークにおけるメッセージ形式を含む、取り合いの差異を吸収し、解法本体の部分の汎用的に用いる実装の枠組みを考慮した。受信、解法の計算、送信の手続きに、解法の処理を分割するモデルとし、各手続きをコールバックとして呼び出すようにした。序盤の検討では、出版-購読型の通信基盤を模擬するシミュレータ上で、エージェントのプログラムを構成した。各コールバックの呼び出し間隔や低 QoS 設定におけるメッセージ欠落を模擬し、各解法の動作を検証した。以降の実通信フレームワーク上での実装においては、メッセージ形式の取り合いを除いて、この段階で作成した解法のプログラムを移植することを基本とし、汎用性を検証した。

MGM は各エージェントが近傍エージェントに同報する。各エージェントを発行者、近傍エージェントを購読者とした。メッセージのトピックは簡単に各エージェントの識別名とし、解法の反復回数や各反復の各段階の共有情報をメッセージのペイロードの一部とした。解法の段階に応じた情報を残りのペイロードに格納した。

AED は MGM と同様に各エージェントは近傍エージェントに同報するとともに、エージェント/変数を頂点とする全域木に従ってメッセージを伝搬する。また、近傍エージェントからの依頼に返信する場合がある。近傍エージェント間の発行者、購読者の通信は MGM と同様とした、近傍エージェント毎に送信するメッセージは、同一のものは括り、異なるものはサイズを抑制した上でパッキングし、同報メッセージに収めた。また、全域木を伝搬するメッセージを、同時に送受されるメッセージにピギーバックさせて統合した、

5-2 解法の実装

ROS2 環境においては、ROS2 ノードをエージェントとした。基本の ROS2/DDS の環境には明示的なブローカはなく、各 ROS2 ノードは任意のトピックで出版・購読する。各エージェントは自身のトピックで出版し、近傍エージェントのトピックについて購読する。一般に、フレームワークが必要とするループ/スピン処理に、解法に関する処理を統合させるか、並行させる必要がある。ここでは、ROS2 の単スレッドモデルのスピン処理を主ループとし、購読のコールバックは ROS2 のスピン処理から呼び出されるようにする一方で、解法の計算と、発行のコールバックは、ROS2 が提供するインターバルタイマーによりそれぞれ呼び出されるように構成した。

メッセージは ROS2 の出版-購読モデルのメッセージ定義に従い、適切なデータ型から構成されるようにした。MGM では同一の整数データ型の領域を、解法の段階に応じて共用する簡易な形式に整理した一方で、AED では個体群を表す可変長配列を含む数種類のデータ領域を、解法の段階に応じて使い分けるようにした。ROS2 ノードのプログラムを C++により記述した。

5-3 検証・評価環境

解法の実機への実装と検証の基礎として、シミュレーションと同様の評価実験を実機上で行う環境が必要となる。実験環境の計算機上への実験設定の配布、エージェント群の起動と実験中の待機、実験終了後の結果データの回収、回収した結果データの統合、の各段階を(半)自動化する必要がある。ここでは、管理用の計算機を別に設け、管理用の計算機から実験用の小型ボード計算機(SBC)群に、一連の指示をするシーケンスを試作した [10]。

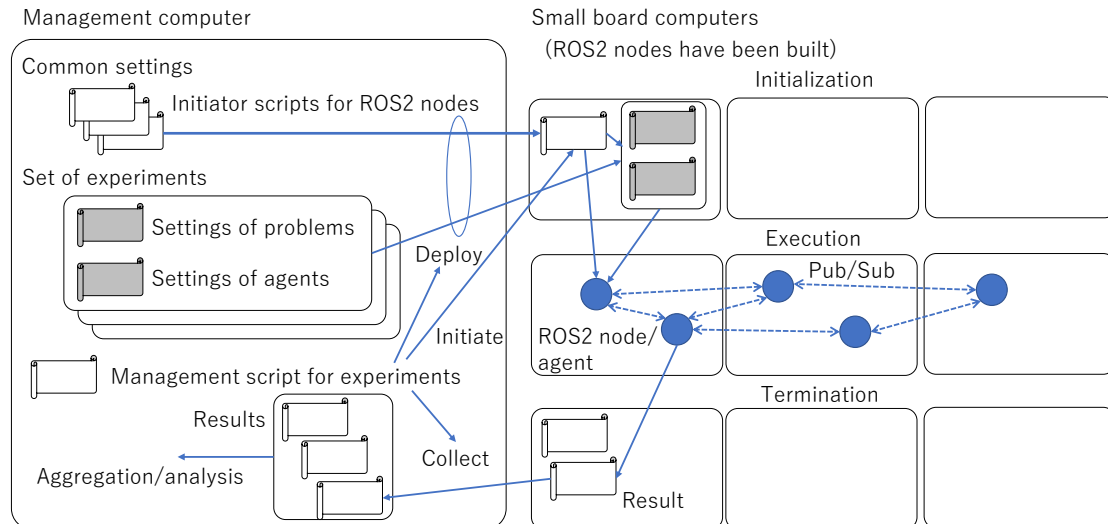


図 2 実験環境の構成 [10]

図 2 に概要を示す．基本的にはシェルスクリプトと SSH を中心に構成し，実験の問題設定，関連するエージェントのパラメータ設定を実験環境の各計算機に転送し，各計算機上で設定に基づき動作する必要数のエージェントを起動し，実験終了後まで待機する．各エージェントのプログラムは事前に各計算機上に配備されているものとし，実験に応じた設定と個数のエージェント群を起動するスクリプトを送り込む．各エージェントは独立して動作し，それぞれの実行結果のファイルを出力する．ここで評価した解法は anytime 手法であるため，各エージェントに実験のカットオフ時間を与え個別に終了させる．実験終了後に，各計算機から各エージェントの実行結果のファイルを回収する．

各エージェントの実行結果は実行開始から終了までの反復処理の各段階の結果のログを含む．エージェントの起動は，分散システム上で順に起動するために必ずしも同期しておらず，反復処理の時間にずれが生じうる．そこで，これらを突き合わせて大域的な解を含む状況の推移を推測する．短期間の実験であることから，簡単のために，エージェントを実行する計算機の持つ同期された時計と，プロセッサのカウンタの情報を用いて，各エージェントの時刻ずれを簡易的に補正して，大域的な anytime 曲線を再構成した．

5-4 結果

小型ボード計算機からなる解法の実験環境(SBC)として，Raspberry Pi 4b, Ubuntu 22.04 LTS, ROS2, g++ の計算機 5 台を用いて，提案手法を実装し，実験により評価した [10] [11]．各 SBC は同一の無線アクセスポイントを介して通信するものとした．

ベンチマーク問題は n エージェント/変数，変数の値域のサイズ 10，目的関数の数を c とした．目的関数の 2 項関数の効用値は，一様分布乱数に基づく $[1, 100]$ の整数および， $(\alpha, \beta) = (9, 2)$ のガンマ分布に基づく $[1, 100]$ の値を丸めた整数とした．

既定の DDS をベストエフォート設定で用いた．AED のパラメータは従来研究 [AED_AAMAS_2020] に従った．従来研究で良好だった設定 (rcv-ub) の MGM と，AED を比較した．AED の最良解コミットのメッセージ損失の対処を，最低限の動作継続 (base)，解品質の局所的な単調性維持 (lcl-mono)，近傍エージェントの最良解も考慮 (nbr-best) とした．各解法のカットオフ期間を 30 秒とした．

表 3 に，最終的な解品質 (utility) と各エージェントが無視した近傍エージェントの平均値 (#ignr.) を示し，図 3 に，起動時からの推移の anytime 曲線の例を示す．実環境における，制御不能なメッセージ損失の頻度とコールバック処理のタイミングのずれが大きく影響し，直接的な比較は難しいが，メッセージ損失が極端でない状況では，各手法は想定に近い動作をした．

表 3 ROS2 環境における解品質 [11]

Prb.	Alg.		Utility	#Ignr.
Random-100 n=5 c=10	MGM	rcv-ub	800.8	0.21
	AED	base	827.8	0.18
		lcl-mono	786.9	0.33
		nbr-best	830.7	0.14
Random-10 n=10 c=25	MGM	rcv-ub	191.9	2.32
	AED	base	196.6	1.44
		lcl-mono	193.8	1.31
		nbr-best	201.7	1.45
Random-100 n=10 c=25	MGM	rcv-ub	1832.4	2.18
	AED	base	1864.3	1.18
		lcl-mono	1852.3	1.23
		nbr-best	1883.1	1.70
Gamma92-100 n=10 c=25	MGM	rcv-ub	575.9	1.94
	AED	base	593.5	1.14
		lcl-mono	583.4	1.2
		nbr-best	592.5	1.1

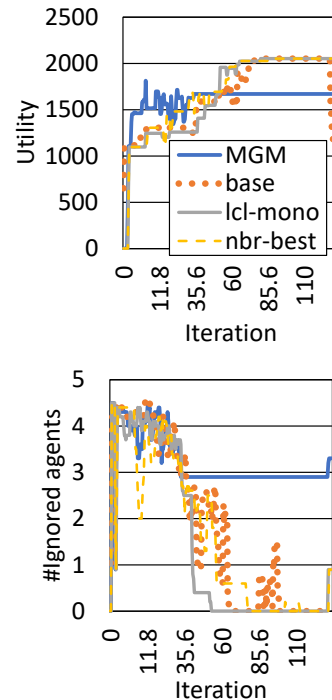


図 3 ROS2 環境における推移 [11]

MGM の立ち上がりと局所最適解への収束の速さと、AED の解品質の優位性もある程度確認できた。

実時間性を考慮する場面では、処理の段階と情報交換が多い AED について、立ち上がり優先や個体数抑制など、解品質と処理資源のトレードオフ調整の余地がある。エージェントの再同期の改良も実装上の課題である。

6 ブローカを伴う環境下での適用

6-1 ブローカを伴う実装モデル

出版者と購読者の間に、明示的なブローカが介在する方式の通信フレームワークを用いる場合は、出版者はブローカに送信し、購読者はブローカから受信する。ブローカは情報の集約、負荷の集中、単一故障点のリスクを伴うノードとなる。協調問題の解法は、ブローカを介する出版-購読型通信網上にオーバーレイ網を埋め込むことになる。情報とリスクの分散のために、複数ブローカを用いる構成における、ブローカの配置なども課題となると考えられる。その基礎として、単一または複数のブローカ構成の系に解法を実装する状況について検討した。

検討の対象とするフレームワークの制限を考慮し、簡単のために、各エージェントは単一ブローカに対してのみ情報を送信した。その一方で、関連するトピックに関わる複数のブローカから購読する構成とした。

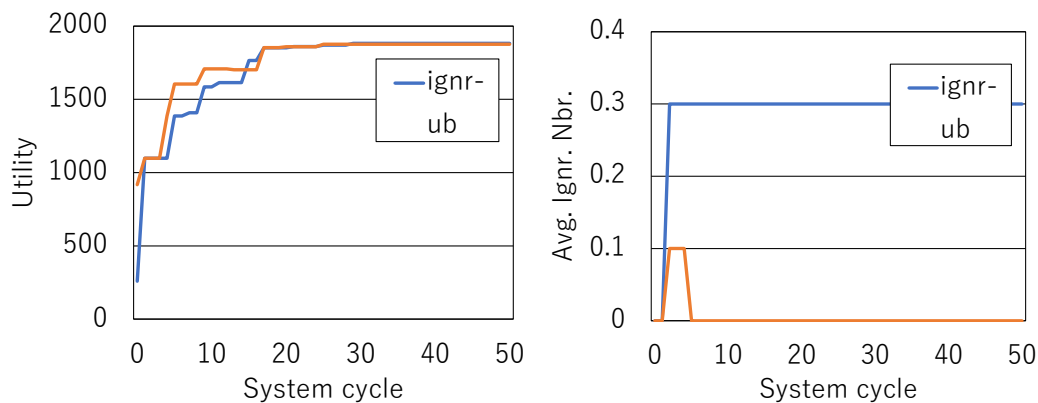
低 QoS 通信を考慮する解法の基本的な構成はこれまでに検討したものを踏襲し、解法の主要部分のプログラムも、取り合いを調整した上でこれまでのものを移植し、実装の汎用性を検証した。

6-2 解法の実装

MQTT 環境である Mosquitto [12] 上に解法を実装した。ここでは、ブローカと発行/購読者のクライアントから系が構成される。複数のブローカを同時に用いることができる一方で、各クライアントは単一のブローカに対してのみ接続できるという実装上の制限がある。そこで、各エージェントは接続すべきブローカに対して、それぞれ MQTT クライアントを並行させる構成とした。

表 4 MQTT 環境における MGM の解品質 [13]

#broker	n, c	alg	Utility	Avg. Ignr. Nbr.
1	5, 10	ignr-ub	799.5	0.11
		rcv-ub	801.3	0.16
	10, 25	ignr-ub	1925.4	0.26
		rcv-ub	1927.6	0.15
5	5, 10	ignr-ub	798.0	0.34
		rcv-ub	817.4	0.13
	10, 25	ignr-ub	1815.2	1.81
		rcv-ub	1918.2	0.43

図 4 MQTT 環境における MGM の推移 ($(n, c)=(10, 25)$, 5 brokers) [13]

各エージェントが持つ各 MQTT クライアントのループ処理はマルチスレッド版を用い、エージェントの主ループと並行させた。各 MQTT クライアントのループから呼び出される購読のコールバックから、解法の購読のコールバックを呼び出すようにした。その一方で、解法の計算と発行のコールバックはそれぞれエージェントの主ループから呼び出すようにし、それぞれのタイミングの比率を調整した。

MQTT のメッセージは任意のバイナリ形式のペイロードを送信できるため、同一環境の計算機間の協調であることを前提に、プログラミング言語の構造型でメッセージを定義しバイナリデータとして送受した。

Mosquitto の開発ライブラリを用いるエージェントのプログラムを C++により記述した。ここでは MGM の実装を先行した。

6-3 検証・評価環境

解法の評価環境は、ROS2/DDS で用いたものと同様のものを用いて、各計算機上でエージェントのノードを起動し実験した。

6-4 結果

小型ボード計算機からなる解法の実験環境(SBC)として、Raspberry PI 5, Ubuntu 24.04 LTS, Mosquitto, g++の計算機 5 台を用いて、提案手法を実装し、実験により評価した [13]。各 SBC は同一の無線アクセスポイントを介して通信するものとした。単一ブローカをある SBC で実行する場合と、各 SBC に単一のブローカを実行する 5 ブローカの場合について実験した。5 エージェントを超える場合は、各 SBC に同数のエージェントを配置した。各 SBC にブローカを配置する場合は、各エージェントは自身の SBC のブローカに対して出版するものとした。

解法 MGM の拡張において、一時的に無視されたエージェントが取りうる変数値を推定する戦略として、本

実験のベンチマーク問題で比較的有効だった、最良値を用いる楽観的な戦略(ub)を用いた。無視されたエージェントを通信回復後に協調に復帰させるか、無視し続けるかの各場合(rcv, ignr)を評価した。各エージェントは起動後のタイミングが同一ではないため、解法の反復回数が異なる場合に再同期の機会を得る規則を設けた。出版と処理の間隔の比を、それぞれ 20, 100 ms とした。購読のタイミングにおいて、2 回以上メッセージが損失したエージェントを、無視することとした。

ベンチマーク問題は n エージェント/変数、変数の値域のサイズ 10, 目的関数の数を c とした。目的関数の 2 項関数は、一様分布乱数に基づく $[1, 100]$ の整数の効用値とした。

表 4 に、解品質と、各エージェントが無視した近傍エージェント数の平均値を示す。外乱の影響がみられ、結果の差異がごく小さい場合があるが、効用値の最大化、無視された近傍エージェント数の削減のいずれにおいても、エージェントの復帰の機会がある場合の方が、平均的には良好な傾向が見られた。ブローカ数が多い方が、ignr-ub と rcv-ub の結果に開きがある。これはノード間をまたがる通信の方向が多様になり増加したことが影響した可能性があると考えられる。

図 4 に、あるインスタンスにおける結果の推移を示す。これらの結果では、欠落したエージェント数が比較的抑制されており、MQTT の QoS 設定は最低の 0 としたが TCP に基づく環境を比較的低負荷で用いたため、通信品質が比較的良好であったと考えられる。特に、各エージェントの起動のタイミングにずれが生じた時に、その回復に協調の復帰が機能する結果となった。これらの結果では ignr-ub がエージェント間の協調を回復していないケースがあるが、立ち上がりを除いて解品質は rcv-ub と同等であった。これは、エージェントの欠落が皆無か小規模であること、局所探索であること、不在のエージェントを補う戦略 ub の効果によるところがあると考えられる。

これらの結果は、小規模な環境と問題においては、比較的安定した挙動が得られたことを示唆していると考えられる。

7 まとめ

ロボット用ソフトウェアプラットフォーム ROS2 や、IoT 機器の通信に用いられる MQTT などの通信フレームワークでの活用の特化した、装置間の分散協調アルゴリズムのソフトウェア基盤技術の開拓を動機付けとして、主要な構成要素である最適化手法の実装技術と有効性について検討した。特に、分散制約最適化問題の解法を実システム上に実装するうえで必要となる、低 QoS 通信におけるメッセージ損失への対処を含む実時間指向の解法、出版-購読型通信モデル上への適用を検討し、シミュレーション、実装、評価環境の試作、実験により検証した。

複数の基礎的な解法を、シミュレーションの段階から、複数の異なる実環境の出版-購読型通信フレームワーク上に実装して、実験環境を整備し検証する段階に発展し、一連の過程により、分散制約最適化手法を実装する上での基礎的な方法論と有効性が確認された。

その一方で、実装作業の時間的コストの制限により、本質部分となる基礎的な解法の実装を中心とした検証となり、解法の準備や出力の分散処理化を含む周辺処理の整備や、ライブラリ化などは今後の課題である。また、各種解法を同様の枠組みで実装できるように移植するにあたっては、異なる解法の処理の一部を類似する方針に基づいて同様に変更する作業を伴うため、その省力化のために AI に基づくプログラムコード生成支援を活用する方法論の模索も課題として挙げられる。

本研究の機会をいただきましたことにお礼申し上げます。

【参考文献】

- [1] "ROS on DDS," [Online]. Available: https://design.ros2.org/articles/ros_on_dds.html. [Accessed 26 6 2025].

- [2] "MQTT: The Standard for IoT Messaging," [Online]. Available: <https://mqtt.org/>. [Accessed 26 6 2025].
- [3] F. Fioretto, E. Pontelli and W. Yeoh, "Distributed Constraint Optimization Problems and Applications: A Survey," *Journal of Artificial Intelligence Research*, vol. 61, pp. 623–698, 2018.
- [4] R. Zivan, H. Yedidsion, S. Okamoto, R. Grinton and K. Sycara, "Distributed constraint optimization for teams of mobile sensing agents," *Autonomous Agents and Multi-Agent Systems*, vol. 29, no. 3, pp. 495–536, 2015.
- [5] F. Fioretto, W. Yeoh and E. Pontelli, "A multiagent system approach to scheduling devices in smart homes," *16th Conference on Autonomous Agents and MultiAgent Systems*, p. 981–989, 2017.
- [6] R. T. Maheswaran, J. P. Pearce and M. Tambe, "Distributed Algorithms for DCOP: A Graphical-Game-Based Approach," *17th International Conference on Parallel and Distributed Computing Systems*, p. 432–439, 2004.
- [7] S. Mahmud, M. Choudhury, M. M. Khan, L. Tran-Thanh and N. R. Jennings, "AED: An Anytime Evolutionary DCOP Algorithm," *19th International Conference on Autonomous Agents and MultiAgent Systems*, p. 825–833, 2020.
- [8] T. Matsui, "An Investigation of Distributed Constraint Optimization with Non-Responding Agents Toward Real-Time Solution Method on Practical Messaging Platforms," *16th International Conference on Agents and Artificial Intelligence*, vol. 1, pp. 428–437, 2024.
- [9] 松井俊浩, "低品質の出版-購読型通信系を用いる異種の分散制約最適化手法の実装と評価のための枠組みの基礎検討," 第 23 回情報科学技術フォーラム, 第 巻 2, pp. 493–495, 2024.
- [10] T. Matsui, "Implementation of Real-Time Local Search Method for Distributed Constraint Optimization with Message Loss in Pub/Sub Communication," *22nd International Conference on Practical applications of Agents and Multi-Agent Systems*, p. 343–349, 2024.
- [11] T. Matsui, "Investigation of Anytime Evolutionary DCOP Algorithm on Pub/Sub Communication Platform with Message Losses," *17th International Conference on Computer and Automation Engineering*, pp. 308–313, 2025.
- [12] "Eclipse Mosquitto (TM) An open source MQTT broker," [Online]. Available: <https://mosquitto.org/>. [Accessed 26 6 2025].
- [13] 松井俊浩, "ブローカを伴う低 QoS の出版-購読型通信を用いる確率的分散制約最適化手法の検討," 第 24 回情報科学技術フォーラム, 2025.

〈発 表 資 料〉

題 名	掲載誌・学会名等	発表年月
出版-購読型通信プラットフォームにおける分散制約最適化手法の基礎検討	電子情報通信学会人工知能と知識処理研究会(AI), vol. 123, num. 190, pp. 157–160	2023 年 9 月
An Investigation of Distributed Constraint Optimization with Non-Responding Agents Toward Real-Time Solution Method on Practical Messaging Platforms	16th International Conference on Agents and Artificial Intelligence, vol. 1, pp. 428–437	2024 年 2 月
分散制約最適化問題の局所探索解法の出版-購読型通信基盤への実装のための基礎検討	情報処理学会第 86 回全国大会, vol. 2, pp. 52–53	2024 年 3 月

メッセージ損失を伴う環境における進化計算に基づく非集中型最適化手法の一検討	2024 年度人工知能学会全国大会 (第 38 回), 2F4-GS-5-01	2024 年 5 月
Implementation of Real-Time Local Search Method for Distributed Constraint Optimization with Message Loss in Pub/Sub Communication	22nd International Conference on Practical applications of Agents and Multi-Agent Systems	2024 年 6 月
低品質の出版-購読型通信系を用いる異種の分散制約最適化手法の実装と評価のための枠組みの基礎検討	第 23 回情報科学技術フォーラム, vol. 2, pp. 493-495	2024 年 9 月
低品質の出版-購読型通信を用いる確率的分散制約最適化手法の比較検討	情報処理学会第 87 回全国大会, vol. 2, pp. 37-38	2025 年 3 月
Investigation of Anytime Evolutionary DCOP Algorithm on Pub/Sub Communication Platform with Message Losses	17th International Conference on Computer and Automation Engineering, pp. 308-313	2025 年 2 月
ブローカを伴う低 QoS の出版-購読型通信を用いる確率的分散制約最適化手法の検討	第 24 回情報科学技術フォーラム	2025 年 9 月 (投稿済, 発表予定)