



Generic agent-based optimization framework to solve combinatorial problems

Faiza Ajmi, Hayfa Zgaya-Biau, Sarah Ben Othman, Slim Hammadi

► To cite this version:

Faiza Ajmi, Hayfa Zgaya-Biau, Sarah Ben Othman, Slim Hammadi. Generic agent-based optimization framework to solve combinatorial problems. IEEE International Conference on Systems, Man, and Cybernetics, Oct 2020, Toronto, Canada. 10.1109/SMC42975.2020.9283316 . hal-02941737v2

HAL Id: hal-02941737

<https://hal.science/hal-02941737v2>

Submitted on 2 Oct 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Generic agent-based optimization framework to solve combinatorial problems

Faiza Ajmi

CRISTAL CNRS UMR 9189

Villeneuve d'Ascq, France

faiza.ajmi@centralelille.fr

Hayfa Zgaya-biau

CRISTAL CNRS UMR 9189

Villeneuve d'Ascq, France

hayfa.zgaya-biau@univ-lille.fr

Sarah Ben Othman

CRISTAL CNRS UMR 9189

Villeneuve d'Ascq, France

sara.ben-othman@centralelille.fr

Slim Hammadi

CRISTAL CNRS UMR 9189

Villeneuve d'Ascq, France

slim.hammadi@centralelille.fr

Abstract—The aim of this paper is to describe our proposed ABOS framework (Agent-Based Optimization Systems) by demonstrating the interest in using the multi-agent approach while operating hybrid metaheuristics to solve Combinatorial Optimization Problems (COP). Two main contributions are highlighted in this work: 1) to show that the alliance of the multi-agent systems (MAS) and the metaheuristics, based on the interaction and the parallelisms concepts, facilitates the hybrid metaheuristics development and allows the simultaneous exploration of different regions of the search space and 2) to demonstrate that the use of the multi-agent approach, in the context of optimization, is a crucial option in the process of hybridization allowing the development of generic structures. These later promote the interaction between metaheuristics independent of the problem to be addressed. Our challenge in this ABOS framework is to endow the participant agents, with a set of rational behaviours allowing them to change in real time their strategies, according to the optimization process evolution. The simulation results show that the collaborative optimization can be effective in some cases, hence the need to set effectively the parameters of the optimization algorithms behaviours and the collaborative protocols. We also demonstrate that the use of ABOS framework with MAS allows a more robust and generic structure, capable with minimal changes handling different COP.

Index Terms—collaborative optimization, multi-agent system, combinatorial optimization problems, hybrid optimization, metaheuristics

I. INTRODUCTION

The collective computational intelligence is a well-known approach to solve Combinatorial Optimization Problems (COP) since it allows the coordination of several algorithms, with a simultaneous exploration of the search space. In this context, several researchers highlight the interest of combining optimization algorithms with distributed coordination methods in order to resolve complex problems [1]. In [2], the authors present an interesting literature review on collaborative techniques to solve optimization problems. The authors highlight the effectiveness to use the multi-agent approach, which is characterized by the distributed representation of the problem or by its simultaneous resolution by a set of interactive entities. The benefit is to take advantage in an effective manner of all the optimization efforts provided by each participating agent. Most research works study the distribution of a problem to several agents so that each agent is responsible of a sub-problem (cooperation). The purpose is to simplify and to

accelerate the resolution of the complex problems. We can quote for example the work of [3] that proposes a solution where an agent continuously adapts itself during the search process using a direct cooperation protocol based on reinforcement learning and pattern matching. We focus in this paper on the simultaneous optimization of the same problem by several agents within a collaboration protocol. In this context, the authors in [4] propose an adaptable architecture called MAGMA (MultiAGent Metaheuristics Architecture) that can resolve the same problem simultaneously by several agents but the collaboration between these agents is decided by another agent. The same system allows, in case of a large neighbourhood search, the problem decomposition into sub-problems, assigned each to an independent agent in a system. In [5], authors apply an hybridization of metaheuristics to feature and member selection within ensemble systems. Authors use a set of agents in order to optimize simultaneously the feature and base classifiers (members) of the ensemble generation process, corresponding to the problem of automatic design of Ensemble Systems. This work is inspired from [6] where the same system is used to solve the TSP. The principle is to resolve simultaneously the problem by a set of particles (agents), each of which has a memory that contains its current position (current solution found), its objective function and the best solution found up to this current position. To move from a position to another, the agent chooses a decision method (to decide where to move) and a learning method (to learn from all that already done) from a list of decision and learning methods also part of its memory. The target position can be abandoned when during the displacement, a better intermediate solution was found. The learning strategy allows the selection, in real time, of the decision method (i.e. metaheuristics like as the Tabu Search algorithm, the Simulated Annealing algorithm) in order to find a better position than the current one. This learning strategy starts with a random decision strategy, which is replaced, as soon as this latter fails after (T) times, by the best strategy found by the other agents of the system. In [6], authors use the particle swarm optimization with a learning strategy for the TSP.

In [7], authors use a multi-agent system where agents are equipped with auto-learning abilities and have the same Genetic Algorithm (GA) behaviour to resolve collectively the VRP.

In this paper, we focus on the resolution of the same COP by a set of autonomous collaborative agents, showing that an additional AI technique like a reinforcement learning method can help the agents to adjust their optimization parameter settings in order to improve the ABOS performances. But, the aim goal of this paper is to describe our ABOS effective collaboration protocol focusing on how agents collaborate in order to converge in the most efficient way possible to the optimal solution. So, we focus in this work on the collaborative technique rather than the reinforcement learning techniques, which will be specified in future works. Concerning works not using additional AI techniques to resolve the same COP; in [8], authors propose a multi-agent flexible architecture (able to solve easily different optimization problems) called MAM for multi-agent architecture for metaheuristics. Each agent in MAM resolves the VRPTW with a GA or with an iterated local search algorithm and variable neighbourhood search metaheuristics. The weak point of this solution is that the agents do not communicate directly between them but within a decision-making center. In [9], the same research team tried to improve its previous system replacing the decision-making center by a an environmental stimulus pool where the agents store or remove the found solutions. The work of [10] is comparable to that of [8] : the agents interact indirectly through an adviser agent. In addition, and basing on the work of [11], the author proposes to resolve optimization problems with high-level hyperheuristic repository of metaheuristics for a better adaptation to the specific features of the problems to resolve. Implemented algorithms within the agent behaviours are: GA, ant colony (ACA), simulated annealing (SAA), tabu search (TSA) but the framework can easily integrate other algorithms like the particle swarm optimization (PSO). In [12], the authors study a distributed computation model that can be used for general resource allocation problems. In the proposed system, the agents share their solutions in order to optimize collaboratively a cost function. The problem with this system is that the number of interactions can explode with increasing the agent number or for large-scale problems. In this paper, and contrary to what has been proposed in the literature, we aim a fully distributed system where the agents are completely autonomous interacting with a collaboration protocol. In [13], authors propose a simple collaboration system between an ant colony optimization agent and a GA agent to resolve the TSP. The authors show that the proposed approach has an effective performance in terms of the quality of solution and the speed of computation. However, no comparison with other interaction configurations has been made (e.g. other interaction protocols). We notice that in the literature, the works that have relied on the multi-agent collaboration for the simultaneous resolution of the same optimization problem are of 2 types: 1) The agents communicate indirectly via a monitoring agent that centralizes the decision-making. 2) The agents communicate directly but in a completely simplistic way: a simple sending of the best solution found from one agent to another: no strategy of collaboration has been put in place. Our goal in this paper is to check whether the implementation of an

evolved collaborative protocol can improve the search for a better solution. So, we focus on the collaboration approach: all the agents collaborate to resolve the same problem adapting their optimization strategies according to their current feat. In this context, the recent publication of [2] points the lack of existing research works that allow the self-turning of the optimization algorithms parameters in order to improve the research strategy especially using multi-agent systems. The goal is to test if the collaboration, in ABOS framework, with different optimization algorithms improves the process of reaching the optimal solution and with which setting plan.

II. THE PROPOSED FRAMEWORK

A. Using the alliance between MAS and metaheuristic to solve the COP

The importance of using MAS in the design of metaheuristic COP is:

- Facilitates the hybridization of metaheuristic algorithms and promotes the interaction between them;
- Make more flexible the implementation of hybrid metaheuristics;
- Allows simultaneous exploration of various regions of the search space.

On the other hand, the use of MAS in the context of the COP promotes the simultaneous exploration of different search space regions, allowing more diversity and finding a set of better solutions rapidly. The approach based on communicated agents is used as a link between various metaheuristic algorithms involved in solving a COP. In this case, each agent controls and performs its own optimization search space and, at the same time exchange information with other agents in the current stage of its search.

B. The multi-agent formalism

In order to model our multi-agent system in a precise and unambiguous way having a robust basis for programming [2], we choose to use the discrete-event formalism draw from the Discrete Event System (DEVS) Specification [14], [15]. This specification is complementary to the graphical representation given by the AUML (Agent UML) modelling [16]. Contrary to what the authors in [14] claim (UML competes with DEVS), we adopt in addition the AUML modelling for a graphical representation (certainly less fine and less precise) complementary to the formal form in order to give a visual aspect to the system. At a given moment (t), we have a set $A = \{A_i\}$ of N agents in our system. Each agent $A_i (i \in [1, N])$ is modelled as an atomic DEVS as follows: $A_i = \langle X, I, Y, S, \delta_{int}, \delta_{ext}, \lambda, ta \rangle$ table I.

C. The intra and inter agent behaviours

The internal agent reasoning. There are 2 types of events: internal and external. The internal events are caused by the internal dynamic of the agent (the intra agent behaviour) while external/exogenous events are independent (the interaction with other agents of the system). An external event can be an input event not generated by the agent or an output event

TABLE I
THE USED FORMALISM SYMBOLS

Symbol	Signification
X	Set of external inputs
I	Set of internal inputs
Y	Set of outputs
S	Set of sequential states
$\lambda : S \rightarrow Y$	Output function
$s \in S$	A sequential state: contains the main information on the status of the agent and especially its condition between two successive events
$ta : S \rightarrow \mathbb{R}_0^+$	The time advance function
Q	The total state set $\{(S, e, DL) s \in S, 0 \leq e \leq ta(S)\}$
$\delta_{int} : Q \times I \rightarrow S$	The internal transition function
$\delta_{ext} : Q \times X \rightarrow S$	The external transition function
e	The time elapsed since the last transition
DL	The Decision law used by the agent to decide its action according to what it perceives in its environment (the offers received)
$q \in Q$	The total state q

representing the reaction of the agent. A state of an agent corresponds to a status based a number of mental components such as: memory, perceptions, commitments, expectations, goals, believes, intentions [17]. In our system, we distinguish the following status:

- The quiescent status (QUIESC) at $t < t_{i0}$, with t_0 the start-up time of the multi-agent system.
- The optimization process status (OPTM).
- The analyses of the current situation status (ACS).
- The collaboration protocol triggering status (NFH).
- The look for the best parameter settings status (BPS).

The state of an agent evolves according of the triggering of an internal or external event through the (internal or external) transition functions (δ_{int} or δ_{ext}). The Fig. 1 shows the dynamic state of the agent when collaborating with the other optimization agents. We notice here that an agent has a cycling behaviour allowing it to be attentive to its environment. This behaviour is like a watchdog: it is asleep if nothing happens and wakes up as soon as the agent perceives anything in its environment (external event). So, in $t < t_{i0}$, an agent is in a quiescent status (QUIESC). So, as soon as the system solicits the agent to resolve a COP, the agent wakes-up, initialises its parameters according to the COP to resolve and then begins its activity at t_{i0} if the event X_{i0} occurs corresponding to the set of inputs of this agent. In our case, X_{i0} contains the COP to resolve and its features. Then within the interval $[t_{i0}, t_{i1}]$, A_i selects its best current solution (cur_sol_i) and then makes transition at t_{i1} (event t_{i1}) to the ACS state knowing that at t_{i1} , it can also receive the best current solutions of the other agents A_j of the system. In $[t_{i1}, t_{i2}]$, the agent A_i , being the ACS status, looks if it improved its current solution computing the gap between the two last best solutions α and β with (the current best solution is α). This gap is noted by ϕ and computed as a stagnation rate belonging to its internal state: $\phi = (\beta - \alpha) / \alpha$. This rate is compared to a threshold parameter noted by ϵ . The next transition occurs at t_{i2} : If $\phi \leq \epsilon$, the agent triggers the NFH collaboration protocol. So, according to the CFH protocol proposed in next section, the help process may take place. In addition, this agent sends its best solution to

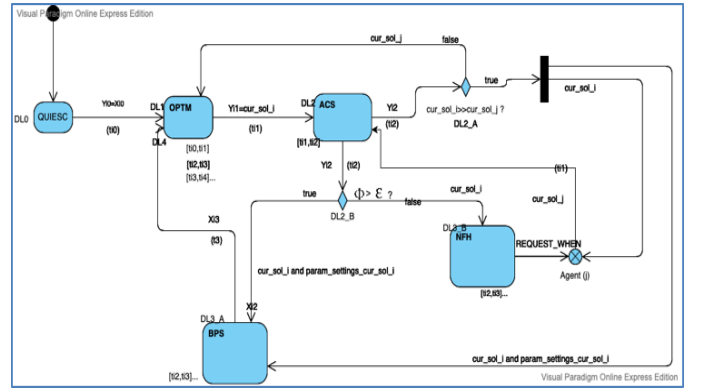


Fig. 1. The internal dynamic of the agent

the agent A_i if it is better than cur_sol_j . Otherwise, A_i injects cur_sol_j in its optimization process. If $\phi > \epsilon$, it means that the agent has (significantly) improved its solution. If the agent A_i improve its solution or if it found better than the agent A_j , it goes to the BPS status to store the current configuration to be used later to the OPTM status looking for the most effective parameter settings allowing it to improve its current solution.

The decision law (DL). The total state $q = (s, e, DL) \in Q$ depends on the status (s), the time (e) elapsed since the last transition and the decision law DL used by the agent to select and rank the offers received from other agents and to decide its action. The dynamic of our collaborative agents is driven by the following decision laws: DL0: The agent wakes-up, identifies the COP with its features and decides to participate to the collaborative optimization process. DL1: the agent triggers the optimization process. DL2_A: the agent sends its best current solution according to a help rate value HR (next section) or considers the received solution in its optimization process. DL2_B: the agent updates the database of the parameter settings or calls for help (NFH protocol). DL3_A: the agent selects the best parameter settings for the next optimization step. DL3_B: the agent triggers the NFH protocol (next section) and DL4: the agent continues the

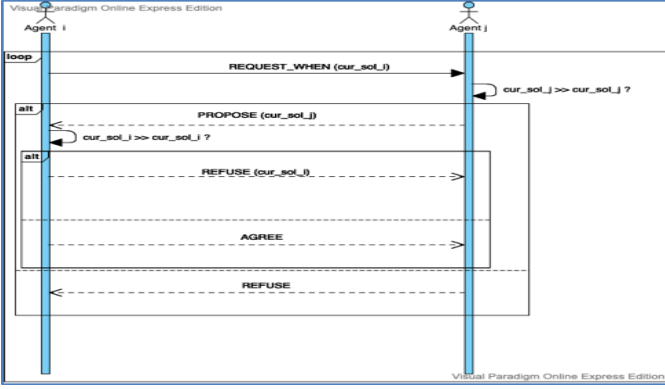


Fig. 2. The proposed NFH protocol

optimization process.

The Need For Help (NFH) interaction Protocol. In MAS domain and in the context of a direct interaction, we call initiator agent, the agent which initiates a conversation and so triggers a protocol. The responder agents are called the participant agents. There are 2 types of messages between agents within a direct interaction: synchronous and asynchronous. A synchronous message blocks the sender agent until it receives a response to its message and an asynchronous message allows the sender agent to continue its work while waiting for an answer to its message. The asynchronous messages are clearly more appropriate to our system allowing the agents to continue to optimize locally their respective current solution while waiting for a response to their request for help. In our MAS, agents interact with the standard FIPA ACL communication language¹ used by the chosen platform for the experimentations (experimentation section). A message between 2 agents A_i and A_j has mandatorily a type, called a performative (e.g. REQUEST, REQUEST_WHEN, REFUSE, AGREE) and can contain information like as the current solution found by the agent. In this section, we propose a collaboration protocol between agents that works as follows (UML sequence diagram of the Fig. 2) ; As soon as the agent A_i decides to ask for help from other agents A_j ($j \neq i$ and $j \in [1, N]$), A_i sends to A_j a REQUEST_WHEN message asking for help if and only if the current message of the agent A_j (cur_sol_j) is better than (symbol $>>$) its own current message (cur_sol_i) transferred as a parameter. If it is the case, the agent A_j proposes its current solution (cur_sol_j) to the agent A_i and refuses otherwise. If the agent A_j proposes its current solution (cur_sol_j) to the agent A_i , this latter checks if its eventually new current solution is better that the received solution since agents continue to optimize while waiting for responses from others (asynchronous messages). If it is the case, the agent refuses the received solution (offering it to the agent A_j) and accepts otherwise (AGREE performative). The decision law (DL2_B) to call to help depends on whether the agent has improved its best current solution after an iteration. So,

we propose in this paper to set up the NFH collaboration, protocol with a Help Rate (HR_j) parameter allowing the agent A_j to help the agent A_i according if a random generated double value $\alpha_j \in [0, 1]$ is less than the $HR_j \in [0, 1]$. The purpose is to check whether the assistance provided really helps the initiator agent A_i . This variable will be used in experimentation section. If $HR_j = 0$ that means that the agent A_j does not collaborate. Other decision laws for this decision step will be studied in future works.

III. EXPERIMENTAL STUDY

For the experimentation, we demonstrate in this paper, as described above, the efficiency of the proposed NFH protocol, in ABOS framework, by resolving the well-known TSP. We choose to use a small MAS composed only by 2 optimizing agents ($N = 2$): the HCAgent having the Hill Climbing optimization algorithm behaviour and the GenAgent having the Genetic optimization algorithm behaviour, varying 4 parameters settings: the number of cities of the TSP, the number of iteration for the HCAgent, the number of generations of the GenAgent and the Help Rate (HR) of the NFH protocol. We compare in this section the MAS functioning with and then without the collaboration using the NFH protocol. In the case of collaboration, we study the performances of the NFH protocol with different values of Help Rate (HR_i) = 1, 0.5 and 0.25 for each agent A_i . We remind here that $HR_i = 0$ means that the agent A_i ($i \in [1, 2]$) does not collaborate and searches for the best solution independently without collaboration.

A. Technical properties of the experimentation

The MAS development platform. For the MAS development, we use the JADE²(Java Agent Development Framework) platform compliant with the ACL-FIPA interaction language. JADE uses java oriented-object language, provides practical packages for the MAS development and offers several graphical tools like the Remote Management Agent (RMA) as the main GUI of the platform, the Introspector agent to debug the agents' behaviours and the sniffer agent to debug conversations between the agents. The Fig.3 shows a scenario of running of our NFH collaboration protocol between the two agents: MyHillClimbingAgent (HCAgent) and MyGeneticAgent (GenAgent) by exploring the content of the REQUEST_WHEN message sent by MyGeneticAgent.

The solution configuration. A solution is a route implemented as an array list of cities (ordered list of cities). A city has a name and is located thanks to its longitude and latitude geographic coordinates. The major met technical problem was to find how the agents can send each other their respective best current solutions in the most effective way possible, with a minimum possible time and avoiding the network congestion, especially when increasing the number of cities and collaborative agents. So, we choose to send current solutions between agents as a string format variable as shown

¹www.fipa.org

²http://jade.tilab.com

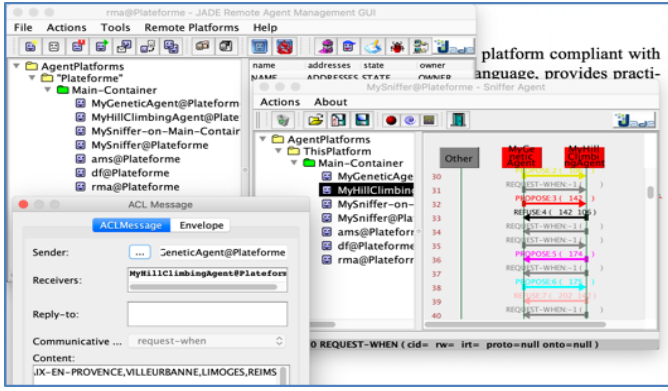


Fig. 3. Jade development platform with graphical tools

in Fig. 3; the agent which receives the current solution of the other agent with a String format, has to transform this received solution as an array list of cities thanks to an SQL request to a MySQL database containing the list of all cities with their geographic coordinates.

The MAS: behaviours and interaction. In Table II, we instantiate the main variables Q, X, I and Y of the formalism of the Table I with: $DL2AB = DL2_A/DL2_B$. The TSP corresponds to the COP with a list of cities, X_{i0} is the initial route, $X_{i1} = \text{cur_sol_}i$ and $\text{cur_sol_}j$; $X_{i2} = \text{cur_sol_}i$ and $\text{param_settings_cur_sol_}i$. We notice here that agents initially receive the same initial route knowing that the COP to resolve is the TSP. But in future work, our challenge is to generalize our system so that the agents will be able to select the most appropriate optimizer behaviour depending on the environment of a special COP. The optimization algorithms are integrated within the agents' behaviours thanks to the `addBehaviour()` method. As mentioned previously, agents can have several behaviours. But we notice here that behaviours within the same agent cannot interact, they can only share common global variables. So, we have a MAS composed of the 2 agents: HCAgent (noted by A_1) and GenAgent (noted by A_2) with the following respective optimization behaviours:

- The HC algorithm is a local search optimization algorithm which starts from an initial solution, generates randomly a neighbour solution thanks to a random transformation of the current solution. This action is reiterated a number of iterations times knowing that the counting restarts from zero each time the current solution is improved. The unique considered parameter of the HC algorithm is the number of iterations that we varied for the experimentation.
- The Gen algorithm is a parallel search optimization algorithm inspired from the biological evolution. This method starts from a set of initial solutions, called population, selects some individual (solutions) to cross (e.g. according the tournament selection) and mute them according to a crossover and mutation rates. Hence the generation of the new population and so on.

In this paper, we focus only on the variation of 3 parameters:

the help rate, the number of iterations for the HCAgent and the number of generations for the GenAgent. The variation of the other parameters and other crossover, mutation and selection operators will be studied in future works. In the following charts (generated in function of the number of cities and the optimization behaviours of the agents), we notice by:

- nbNC: the number of iterations of the HCAgent or the number of generations for the GenAgent without collaboration: $HR_i = 0$ for A_i (NC case). We have chosen 200NC.
- nbNFH: the number of iterations of the HCAgent and the number of generations for the GenAgent corresponding to the collaboration case with the NFH protocol varying the help rate HR_i to : 1, 0.5 and 0.25 for agent A_i . So, we chosen 200 iterations/generations.
- WPS: corresponds to the parameter settings variation (With Parameter Settings).

In our ABOS framework, the collaboration is not always helpful for both agents because it depends to the efficiency of the parameter settings. For example, according to the simulations, for the number of cities = 50 and number of iterations = number of generations=50, the collaboration is not helpful but it is the case for a number of generations =100 (100NFH with $HR_1 = HR_2 = 0, 25$). Hence, the advantage of the help rate proposed in this paper. The collaboration begins to be really interesting for a number of cities = 100 with 50NFH and $HR_i = 1$ for both agents A_1 and A_2 which corresponds to the highest collaboration rate possible. We observe additionally that increasing the number of iterations/generations of the agents does not always improve the results. However, for a number of cities = 200 (Fig. 4 and 5), an average rate of collaboration improves the results $HR_i = 0, 5$ for both agents A_i . That's why the best way is to adjust parameter settings according to the needs of the agents. For example, for 200 cities and 50 iterations, the HCAgent has to collaborate firstly with $HR_1 = 0.25$ then it has to stop the collaboration during 0,048ms, then it has to restart the collaboration at 0,11ms (since the beginning of the simulation) and increase the collaboration ($HR_1 = 1$) at 2,27ms since the beginning of the simulation and so on. In this paper, we made this adjustment parameter settings manually for the purposes of the simulations in order to demonstrate the advantages of this adjustments. But the goal in future works is to develop a learning system that changes automatically the parameter settings in real time as proposed in the internal agent reasoning section.

B. Comparative analysis with other frameworks

In this section, we present a comparative analysis of our framework ABOS with other based-agent frameworks cited in [2]: AgE, AMAM, AMF, CMA, DAFO, JABAT, LBMAS, MACS, MAGMA and MANGO. The Table III summarizes the comparison analysis of several characteristics of multi-agent frameworks including our ABOS framework. The based agent frameworks, such as AgE, AMAM, AMF, CMO, DAFO, JABAT, LBMAS, MACS, MAGMA and MANGO implement

TABLE II
VARIABLES INSTANTIATION

Q			X	I	Y
s	e	DL			
QUIESC	$t < t_{i0}$	DL0	TSP	N/A	Y_{i0}
OPTM	t_{i0}	DL1	X_{i0}	I_{i1}	Y_{i1}
ACS	t_{i1}	DL2AB	X_{i1}	$I_{i2} = \phi$	Y_{i2}
BPS	t_{i2}	DL3_A	X_{i2}	I_{i3}	Y_{i3}
NFH	t_{i2}	DL3_B	X_{i2}	$I_{i3} = \phi$	Y_{i3}
OPTM	t_{i3}	DL4	X_{i3}	I_{i4}	Y_{i4}

TABLE III
COMPARISON OF MULTI-AGENT FRAMEWORKS CHARACTERISTICS

Framework	Described Environment	Autonomy	Synchronous/Asynchronous	Collaboration/Cooperation
ABOS	Yes	Yes	Asynchronous	Yes/No
AgE	Yes	Yes	Asynchronous	No/Yes
AMAM	Yes	Yes	Asynchronous	Yes/No
AMF	Not clear	Yes	Asynchronous	No/Yes
CMA	Not clear	Yes	Asynchronous	Yes/No
DAFO	Yes	Yes	Synchronous	Yes/Yes
JABAT	Not clear	Yes	Asynchronous	Yes/Yes
LBMAS	Not clear	Yes, limited	Asynchronous	Yes/No
MACS	Not clear	Yes	Synchronous	Yes/No
MAGMA	Yes	No	Asynchronous	No/Yes
MANGO	Not clear	Yes	Asynchronous	Yes/No

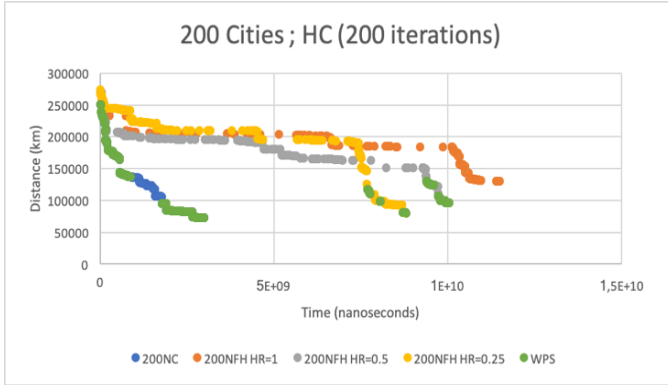


Fig. 4. HCAgent – 200 iterations - 200 cities

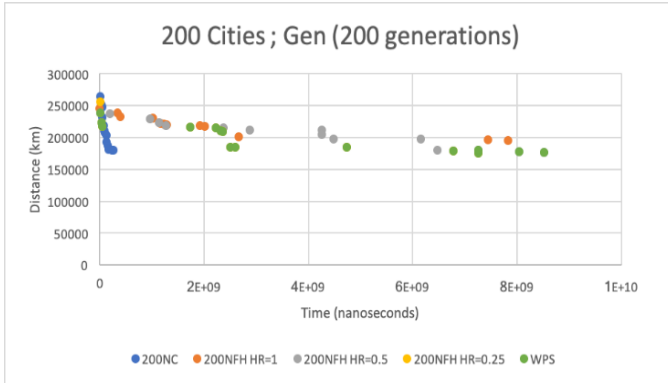


Fig. 5. GenAgent – 200 generations - 200 cities

hyper-heuristics, in an indirect manner, by integrating agent control and coordination. Firstly, most framework environments, in which the agents evolve, are not well described (not clear). The environments of ABOS, AgE and AMAM correspond to the search space of the concerned problem. The environment of DAFO framework is defined by the optimization problem, which is specified by the user. The MAGMA environment is defined by a triple of variables corresponding to the set of solutions, the neighbourhood structure and an objective function. The autonomy of an agent corresponds to its ability to make its own decisions. In LBMAS framework, the agents are not completely autonomous while in MAGMA framework the agents are completely dependent to execute their tasks. For the other cited frameworks like ours, the agents are completely autonomous. In fact, an agent in ABOS does not rely at all on the other agents; it assumes that they represent a potential and not mandatory support. So, an ABOS agent makes the maximum effort on its own to achieve a better solution than its current solution and sends a request for help as soon as it detects a beginning of stagnation without stopping looking its optimization process. Moreover, as soon as the requested agent replies, the ABOS agent compares its last best solution to the received one: if the latter is better, the ABOS agent takes it into consideration to improve its current best solution, otherwise it rejects this received solution and so on. Concerning the synchronous/asynchronous characteristic, an asynchronous agent is able to wait for an answer to its request, ongoing its optimization work like it is the case of our ABOS framework: an agent ask for the help but continues to optimize the COP thanks to parallel behaviours. This technique is very interesting in the sense that the exploration of the research

space does not stop under any circumstances, accelerating the optimization process. Concerning the collaboration or the cooperation behaviour of the MAS, both are interesting and the choice depends of COP characteristics like as the possibility to divide the problem into several interactive subproblems or the complexity of forming the global solution from several complementary subsolutions. The AgE, AMF, DAFO, JABAT and MAGMA Frameworks in Table III use the cooperation concept to find the global optimal solution. Indeed, the concerning frameworks use different agents which working together by exchanging specific information in order to reach a common goal. In MAGMA architecture, the agents are autonomous in the control of their specific activities. The cooperation in MAGMA depends on its architecture levels structure. In ABOS, we focus on the collaboration between optimization algorithms embedded in agents' behaviours in order to promote autonomy, interaction and behavioural parallelism. This collaboration is used as a strategy for the exchange of information between the algorithms involved in finding the solution of the problem. In this context, the exchange of information has, as the main purpose, to guide the search for the most promising regions of the solution space. In turn, parallelism allows simultaneous execution of methods and the consequent reduction on search time. The combination between collaboration and parallelism, which is the adopted strategy in ABOS, is being intensively developed and is becoming each day more important in the optimization context. The frameworks mentioned above have autonomous agents and their cooperation process do not interfere in this autonomy. These characteristics are crucial for the future developments and expansion of those frameworks including of course the ABOS framework.

IV. CONCLUSION AND PROSPECTS

In this paper, we have demonstrated the advantages of the multiagent collaboration to solve a COP thanks to the proposed of ABOS framework using the NFH protocol, knowing that no existing framework in the literature demonstrates this interest with a real direct communication between a set of autonomous entities. We have also demonstrated that this collaboration is not always helpful, so we proposed in this paper to dose it with a help rate. We propose many perspectives to this work: firstly, we have to analyse the proposed protocol varying more parameter settings in order to study more finely the collaboration, we have also to try other optimization behaviours and obviously increasing the number of collaborative agents, using a the most less time-consuming alternative for the communication. We have also to integrate the learning technique in our ABOS framework for the self-turning the parameter settings. Finally, our challenge in long term is to generalize our ABOS framework so that the agents will be able to select the most appropriate optimizer behaviour for a special COP.

REFERENCES

- [1] Wang, Dong and Wang, Wei and Liu, Yurong and Alsaadi, Fuad E, "A modified distributed optimization method for both continuous-time and discrete-time multi-agent systems", *Neurocomputing*, 725–732, 2018.
- [2] Silva, Maria Amélia Lopes and de Souza, Sergio Ricardo and Souza, Marcone Jamilson Freitas and de Franca Filho, Moacir Felizardo, "Hybrid metaheuristics and multi-agent systems for solving optimization problems: A review of frameworks and a comparative analysis", *Applied Soft Computing*, 433–459, 2018.
- [3] Martin, Simon and Ouelhadj, Djamila and Beullens, Patrick and Ozcan, Ender and Juan, Angel A and Burke, Edmund K, "A multi-agent based cooperative approach to scheduling and routing", *European Journal of Operational Research*, 169–178, 2016.
- [4] Milano, Michela and Roli, Andrea, "MAGMA: a multiagent architecture for metaheuristics", *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 925–941, 2004.
- [5] Neto, Antonino A Feitosa and Canuto, Anne MP and Xavier, João C and Barreto, Cephas A, "A multi-agent metaheuristic hybridization to the automatic design of ensemble systems", 2017 International Joint Conference on Neural Networks (IJCNN), 3967–3974, 2017.
- [6] Souza, Givanaldo R and Goldbarg, Elizabeth FG and Goldbarg, Marco C and Canuto, Anne MP, "A multiagent approach for metaheuristics hybridization applied to the traveling salesman problem", 2012 Brazilian Symposium on Neural Networks, 208–213, 2012.
- [7] Silva, Maria Amélia Lopes and de Souza, Sérgio Ricardo and Souza, Marcone Jamilson Freitas and de Oliveira, Sabrina Moreira "A multi-agent metaheuristic optimization framework with cooperation", 2015 Brazilian Conference on Intelligent Systems (BRACIS), 104–109, 2015.
- [8] Fernandes, Filipe Costa and de Souza, Sérgio Ricardo and Silva, Maria Amélia Lopes and Borges, Henrique Elias and Ribeiro, Fábio Fernandes "A multiagent architecture for solving combinatorial optimization problems through metaheuristics", 2009 IEEE International Conference on Systems, Man and Cybernetics, 3071–3076, 2009.
- [9] Silva, Maria Amélia Lopes and de Souza, Sérgio Ricardo and de Oliveira, Sabrina Moreira and Souza, Marcone Jamilson Freitas "An agent-based metaheuristic approach applied to the vehicle routing problem with time-windows", *Proc. of the 2014 Brazilian Conference on Intelligent Systems-Enc. Nac. de Inteligência Artificial e Computacional (BRACIS-ENIAC 2014)*, 2014.
- [10] Malek, Richard "An agent-based hyper-heuristic approach to combinatorial optimization problems", 2010 IEEE International Conference on Intelligent Computing and Intelligent Systems, 428–434, 2010.
- [11] Ouelhadj, Djamila and Petrovic, Sanja and Ozcan, Ender "A multi-level search framework for asynchronous cooperation of multiple hyper-heuristics", *Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference: Late Breaking Paper*, 2193–2196, 2009.
- [12] Nedic, Angelia and Ozdaglar, Asuman "Distributed subgradient methods for multi-agent optimization", *IEEE Transactions on Automatic Control*, 48–61, 2009.
- [13] Tiejun, Zhou and Yihong, Tan and Lining, Xing "A multi-agent approach for solving traveling salesman problem", *Wuhan University Journal of Natural Sciences*, 1104–1108, 2006.
- [14] Zeigler, Bernard P and Praehofer, H and Kim, TG "Theory of Modeling and Simulation: Integrating Discrete Event and Continuous Complex Dynamic Systems, ed, 2000", Academic press.
- [15] Bauer, Bernhard and Müller, Jörg P and Odell, James "Agent UML: A formalism for specifying multiagent software systems", *International journal of software engineering and knowledge engineering*, 207–230, 2001.
- [16] Wagner, Gerd "The Agent–Object–Relationship metamodel: towards a unified view of state and behavior", *Information Systems*, 475–504, 2003.
- [17] Juneja, Dimple and Jagga, Ankit and Singh, A "A review of FIPA standardized agent communication language and interaction protocols", *Journal of Network Communications and Emerging Technologies*, 179–191, 2015.