An Organizational View of Metaheuristics

David Meignan
Systems and Transportation Laboratory
University of Technology of Belfort-Montbéliard
Belfort, France
david.meignan@utbm.fr

Jean-Charles Créput
Systems and Transportation Laboratory
University of Technology of Belfort-Montbéliard
Belfort, France
jean-charles.creput@utbm.fr

Abderrafiâa Koukam
Systems and Transportation Laboratory
University of Technology of Belfort-Montbéliard
Belfort, France
abder.koukam@utbm.fr

ABSTRACT
This paper presents AMF, an Agent Metaheuristic Framework that aims at supporting the design and hybridization of metaheuristics. The introduction of an agent-oriented approach allows to deal with flexibility, robustness and modularity in metaheuristics. This framework is based on an organizational model which describes a metaheuristic in terms of roles. These roles correspond to the main components or tasks in a metaheuristic: intensification, diversification, memory and adaptation or self-adaptation. Starting from this organizational model of metaheuristic, some guidelines allow to obtain a multiagent system that correspond to a particular metaheuristic. In addition, we introduce an original metaheuristic called Coalition-Based Metaheuristic (CBM) to illustrate the use of AMF. Efficiency of CBM is illustrated thanks to its application to the Vehicle Routing Problem.

Categories and Subject Descriptors
I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search—*Heuristic methods*; I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Multiagent systems*

General Terms
Design

Keywords
combinatorial optimization, metaheuristic, multiagent system

1. INTRODUCTION
Research in metaheuristics has recently evolved to new issues. It concerned not only the metaheuristics performances facing dynamic and large problems instances, but it was also interested in proposing simple, flexible, robust and modular metaheuristics. These features have been put forward in several articles and surveys [22, 2]. Simplicity, flexibility, robustness and modularity are subjective but they constitute important criteria for an effective use of metaheuristics.

Distributed Artificial Intelligence (DAI) and particularly multiagent systems seem to be a promising field of research to tackle these new issues. Multiagent approach is tightly linked to metaheuristics considering that both approaches exploit the social metaphor and self-organization paradigm. Thus, the multiagent approach is widely used in metaheuristics, particularly in population-based, hybrid and distributed metaheuristics.

Integration of DAI components in metaheuristics suffer from a lack of tools. Thus, in this paper we propose AMF, an Agent Metaheuristic Framework that aims at supporting the design and hybridization of metaheuristics using an agent-oriented approach. This framework is based on an organizational model which describes the metaheuristics in terms of roles and interactions. A particular metaheuristic can be viewed as a refinement of this model.

This paper also introduces an original metaheuristic called Coalition-Based Metaheuristic (CBM) that illustrates the use of AMF. This metaheuristic is then applied to solve the Vehicle Routing Problem (VRP). CBM combines classical metaheuristic approach and DAI concepts. In CBM, several agents organized in a coalition treat simultaneously an optimization problem. These agents cooperate to perform a better search of solutions. The cooperation consists in exchanging information about the search space and sharing of experiences to improve the agents behavior. The main features of this approach are (i) the use of an agent-based decision process, (ii) the introduction of unsupervised learning mechanisms and (iii) the exploitation of cooperation between agents.

The paper is organized as follows. Section 2 gives a brief overview of adaptive memory programming scheme before introducing AMF in section 3 . Section 4 and 5 present CBM and report experiments carried out on the VRP. Finally, section 6 provides some conclusion statements.

2. BACKGROUND
Multiagent concepts are widely used in metaheuristics, particularly in population-based, hybrid and distributed metaheuristics. The advantages of using multiagent and organi-

zational approaches for metaheuristics design may be justified by, the distribution and robustness inherent to multiagent systems [20], and the need of flexibility and modularity [15]. Our objective in this paper is to propose an organizational and multiagent framework to design and hybridize metaheuristics. The starting point of the study is the Adaptive Memory Programming (AMP) [19] approach which provides the basic concepts of metaheuristics.

AMP is a global scheme that aims at unifying several metaheuristics concepts. Adaptive memory programming has been introduced by Glover in [7], to define the strategic memory components in metaheuristics which guide the intensification and diversification processes. The concept has been extended in [19] to produce an unified view of metaheuristics. In this scheme, a metaheuristic can be viewed as an iterative process summarized in algorithm 1.

Algorithm 1 AMP algorithm scheme

```
Initialize the memory
while stopping criterion is not reached do
    Generate a new provisional solution s using data stored
    in the memory.
    Improve s by a local search; let s' be the improved
    solution.
    Update the memory using the pieces of knowledge
    brought by s'.
end while
```

The AMP approach put forward the iterative process common to several metaheuristics and the concept of memory that supports the search. These elements are the basic concepts of our proposed framework. Since AMP is too general to be used as a framework to design metaheuristics [15] we propose AMF: an Agent Metaheuristic Framework. This framework is based on an organizational model which describes a metaheuristic in terms of roles. The AMF model extends the AMP scheme by adding the concepts of *intensification*, *diversification* and *adaptation* while keeping a high level of abstraction. In addition, we propose an iterative specification process to define a metaheuristic using multiagent systems.

3. AN AGENT METAHEURISTIC FRAMEWORK

This section introduces an Agent Metaheuristic Framework called AMF. It aims at supporting the design and hybridization of metaheuristics. We first describe RIO (Role Interaction Organization) meta-model which ensures the description of the organizational model, then we detail the main components of the organizational model of metaheuristic. In section 3.3, we depict some metaheuristics with the different concepts introduced. Finally, the methodological guidelines that support the specialization process are given.

3.1 The RIO meta-model

By considering organizations as blueprints that can be used to define a solution to a problem, we believe that an organizational approach encourages a reusable model. Thus, to describe a metaheuristic we use an organizational approach based on the RIO meta-model [10]. RIO introduces three basic concepts: Role, Interaction and Organization. A role

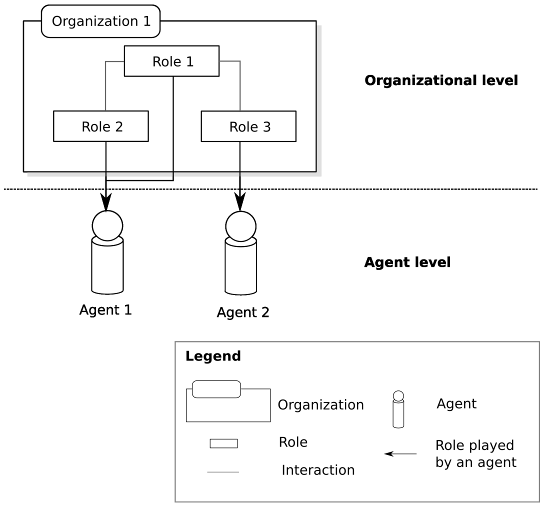

Figure 1: RIO model example

is an abstraction of a behavior or a status in an organization. An interaction links two roles in a way that an action in the first role produces a reaction in the second. An organization is defined by a set of roles and their interactions. These three elements allow to describe a system without making any assumption on the entity which plays the different roles. From these concepts an agent is specified as an active communicative entity which plays roles. An agent may be associated to one or more roles and a role may be played by one or more agents. An example of RIO diagram is presented in figure 1. At the organizational level an organization composed of three roles is depicted. At the agent level the associations of roles to agents are specified.

The proposed metaheuristic framework is based on an organizational model of metaheuristics. This model describes a metaheuristic in terms of organization, roles and interactions. In the following section we describe how RIO metamodel is used to provide an organizational view of metaheuristics.

3.2 The metaheuristic organization

From a multi-agent point of view we define a metaheuristic as an organization. The goal of this organization is to efficiently explore the search space in order to find nearoptimal solutions. This exploration combines intensification and diversification tendencies. To guide the exploration and balance these two tendencies, structured information about the search space is used by subordinate procedures as heuristics. In addition, the strategies used to guide, intensify and diversify may be adapted according to search experiences. Four roles stems from this definition: intensifier, diversifier, guide and strategist. The resulting metaheuristic's abstract organization is represented in figure 2. The definitions of these four roles are given below.

3.2.1 *Intensifier and diversifier roles*

Intensification and diversification are the fundamental tasks in metaheuristics. We assign a specific role to these tasks, respectively intensifier and diversifier. These two roles correspond to the search task in AMP approach. On the contrary

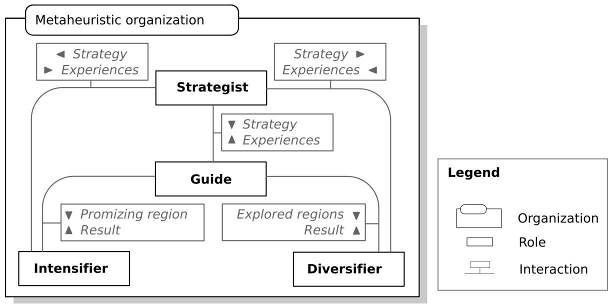

Figure 2: Organizational view of metaheuristics

to the AMP, intensification and diversification are considered separately.

The goal of the intensifier role is to perform a search in a promising region of the search space. To restrict the search area, the intensifier uses information such as a starting solution or constraints on the search space. The goal of the diversifier role is to identify new promising regions in the search space. The diversifier possibly uses information about already explored regions to perform its task.

These two roles can refer to a single process or two distinct ones. For instance, in a basic tabu search, the search is only performed by a local search. However, in the iterated local search, intensification and diversification correspond respectively to a local search and a perturbation [2].

3.2.2 Guide role

The goal of the guide role is to balance diversification and intensification tendencies, and to coordinate diversifier and intensifier roles. The guide role is an intermediate between intensifier and diversifier roles. It structures the information obtained by these two last roles and leads the search.

The guide role corresponds to the management of the memory in AMP scheme. The memory managed by the guide role and the information provided to intensifier and diversifier roles can take several forms. For instance, in tabu search the memory is composed of a tabu list; in evolutionary algorithms, the memory is constituted by a population of solutions; in ant colony algorithms, the pheromone trail may be considered as a kind of memory. Thus, the concept of "memory" is the base of the guide role.

3.2.3 Strategist role

Several hybrid metaheuristics introduce adaptation mechanisms. The term "adaptation" denotes a modification of the search strategy according to the optimization context. This context is defined by the studied problem and the current state of the optimization process. In the organizational model, the adaptation task is assigned to the strategist role. Its goal is to adjust or change the strategies of guide, diversifier and intensifier roles according to the context.

The adaptation in metaheuristics is a process that aims at improving the performance of the search or reducing the parameter setting. Thus, adaptation mechanisms related to the strategist role, adjust or change the search strategies to suit the problem instance or the state of the optimization process. For instance, in genetic algorithms, a population sizing scheme is a task related to the strategist role in our organizational model. This particular role is not investigated in the AMP scheme.

3.3 An organizational view of metaheuristics

In the previous part, we presented an organizational model of metaheuristic based on four roles. This model can be considered as an unified view of several metaheuristics and as an extension of the adaptive memory programming approach. The defined organizational model presents abstract concepts that must be refined according to the problem to be solved. Table 1 attempts to present the components of metaheuristics associated to the roles for several representative metaheuristics: Tabu Search (TS), Simulated Annealing (SA), Genetic Algorithms (GA), Ant Colony Optimization (ACO), Iterated Local Search (ILS) and Variable Neighborhood Search (VNS). The terms used to characterize the components of metaheuristics draw from [2, 8].

The two first columns present the components of metaheuristics related to the effective search of solutions or partial solutions. Intensification and diversification processes can be distinguished for several metaheuristics as GA, ILS and VNS.

The components related to the guide role, in the third column, use intensifier and diversifier roles to manage the search. The guide corresponds to the main strategic components that allow to balance the intensification and diversification tendencies. The memory managed by the guide role takes several forms: a tabu list in TS; a single solution in SA, ILS and VNS; a population of solutions in GA; or a pheromone trail in ACO. The main tasks of the guide role, already identified in AMP scheme, consist of (i) updating the memory and (ii) providing informations to ensure the construction of new solutions, partial solutions or population of solutions.

In the last column of table 1 some components of adaptive or self-adaptive extentions of metaheuristics are presented. This description of adaptive components is not exhaustive. For the tabu search metaheuristic, the Reactive Tabu Search (RTS) approach [1] introduces a mechanism to adapt the tabu list size. In the case of simulated annealing metaheuristic, the Adaptive Simulated Annealing (ASA) approach [11] allows to adapt the temperature schedule. Several adaptive extensions of evolutionary algorithms exist but we have retain the three major issues described in [14] that adapt the mutation rates, recombination probabilities and the population size. These features related to the strategist role modify the search strategy by adjusting the setting of control parameters. This table shows that our framework can model a wide range of metaheuristics.

3.4 Methodological guidelines

To obtain a metaheuristic from the AMF organizational model, it is necessary to refine the different roles and determine the multiagent structure of the optimization system. Thus we provide some methodological guidelines to assist the design of a particular metaheuristic starting from AMF organizational model. The result of this design process is a multiagent system that correspond to a metaheuristic. This

Table 1: Realization of AMF roles for representative metaheuristics

Metaheuristic	Intensifier	Diversifier	Guide	Strategist
TS	Neighborhood search		Tabu list update; Current solution choice	Adaptation of tabu list size (RTS)
SA	Neighborhood move		Current solution choice with acceptance criterion; Temperature update	Adaptation of temperature schedule (ASA)
GA	Recombination	Mutation	Selection	Adaptation of: population size, mutation rates, recombination probabilities
ACO	Solution construction		Pheromone update	
ILS	Local Search	Perturbation	Current solution choice with acceptance criterion	
VNS	Local Search	Shaking	Current solution choice; Neighborhood choice	

process draws from RIO methodology [9], and is composed of three phases:

1. AMF Roles refinement
2. Agentification
3. Metaheuristic specialization

The first phases consists in determining the means that are required to perform the different roles described in the AMF organizational model. Thus, it is necessary to analyze the optimization problem characteristics (difficulty, known optimization methods, etc.) as well as the functional and non-functional requirements (robustness, distribution, etc.). The result of this phase consists in a particular organizational model of a metaheuristic with a description of each role behavior and interaction.

The agentification allows to determine the multi-agent structure of the metaheuristic. It consists of (i) the identification of the different types of agents, (ii) the assignment of roles to these agents and (iii) the description of roles scheduling for each type of agent. This phase allows to determine a multiagent system able to solve one or several class of optimization problems.

The final phase consists in specializing the multiagent system to treat a particular optimization problem. For instance, if a genetic algorithm has been described in the previous phase, the specialization consists in the determination of mutation and recombination operators considering a particular optimization problem.

4. A COALITION-BASED METAHEURISTIC

This section illustrates the AMF methodological guidelines with the design of an original metaheuristic called Coalition-Based Metaheuristics (CBM). In the section 4.1, the main features of CBM are detailed. Then, we follow the guidelines of our framework to describe CBM. Finally, the last two sections put the emphasis on the decision process and learning process related to a CBM agent.

4.1 CBM principle

CBM is a metaheuristic based on the metaphor of a coalition. The term "coalition", drawn from [16], designates a multiagent system where agents have the same capacities and cooperate by means of direct interactions. In our case, the coalition is composed of several agents which have the capacity to individually treat the optimization problem but cooperate to coordinate and improve the search.

To perform the search a CBM agent manages a single solution and uses several operators to move in the search space. These operators are related to the intensification task or diversification task. Intensification operators refer to improvement processes such as local search procedures, and diversification operators correspond to generation, mutation or crossover procedures. The search procedure involved by an agent have some similarities with Variable Neighborhood Search (VNS), (see for instance [8]). A set of local search operators are applied on the current solution until a local minimum is reached, then a perturbation is performed using a diversification operator. The schedule of the operators is determined by a decision process. This strategic process selects the most appropriate operator considering the optimization context. In our case, the optimization context refers to the problem instance and the evolution of the optimization process. The search behavior of an agent is adapted during the optimization by learning mechanisms. These learning mechanisms modify the rules of the decision process according to the search experiences.

In addition, agents can cooperate by two ways. In one hand, an agent shares its best known solution. This solution can be exploited by other agents with crossover operators. This cooperation is intended to guide the search through new promising region of the search space. In another hand, an agent can share its internal decision rules in order to allow mimetism of behavior. This second cooperation mechanism is intended to favor the search behaviors that often found new best solutions.

The main advantages of the CBM are the flexibility and the robustness. The flexibility is the capacity to solve various type of problem, and the robustness is the ability to maintain performance in a perturbed environment. To tackle these two points we propose a distributed metaheuristic where several subordinate heuristics are used. The distribution of roles over several agents guarantee the robustness. Thus, addition or removal of an agent does not perturb the global functioning of the system. Combination of different subordinate heuristics and the use of simple adaptive mechanisms ensure effectiveness of the optimization. CBM is particularly designed for (i) combinatorial optimization problems where several neighborhood structures can be exploited and (ii) problems that require a high computational capacity which justify a distribution.

4.2 CBM Roles refinement

To describe the main components of CBM, we first refine the AMF organizational model and detail the behavior of each roles. Figure 3 presents the CBM organization resulting from refinement of AMF organizational model.

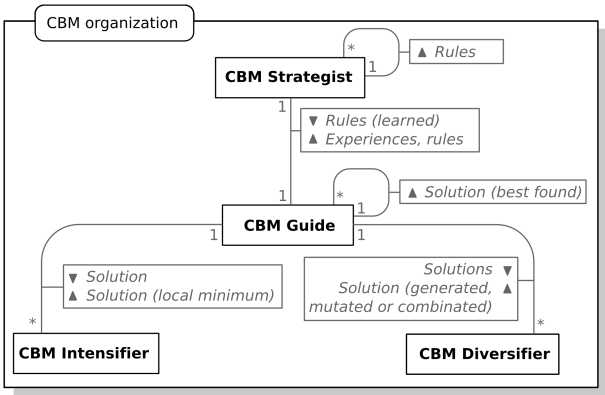

Figure 3: AMF organizational model refinement for CBM

Intensifier and diversifier roles: In the coalition-based metaheuristic, the intensifier and diversifier roles are related to several operators. Intensification operators refer to local search processes and diversification operators correspond to generation, mutation or crossover operators. Thus, *CBM intensifier* and *CBM diversifier* role behaviors are simple tasks which provide solutions to the *CBM guide role*.

Guide role: The *CBM guide role* manages a set of three solutions, the current solution $s_{current}$, the best found solution $s_{bestfound}$ and the best known solution $s_{bestknown}$ obtained by interaction with others guide roles. The *CBM guide role* coordinates the intensifier and diversifier roles thanks to a decision process that enable the choice of the operator to apply according to the optimization context. This decision process is detailed in the following section. In addition, the *CBM guide role* shares its best found solution. This solution can be exploited by crossover operators managed by the diversifier role.

Strategist role: The *CBM strategist role* aims at adapting the decision process rules used by the *CBM guide role*. This adaptation is performed by reinforcement learning and mimetism learning. The reinforcement learning consists in observing the search experiences and favoring the rules that lead to good solutions. This learning is performed by an interaction with a single *CBM guide role*. The mimetism learning consists in copying a part of the decision rules coming from others *CBM strategist role* and identified as efficient. These two learning mechanisms are presented in the next sections.

4.3 Agentification

The refined model of CBM allows to describe the agents and the structure of the organization. In the coalition, an agent plays all the roles: *CBM Intensifier*, *CBM Diversifier*, *CBM Guide* and *CBM Strategist*. The CBM system is composed of a fixed number of identical agents. These agents are organized in a coalition where each agent can interact with all other agents. The cooperation between agents occurs for the guide role and the strategist role. The agentification is illustrated in figure 4.

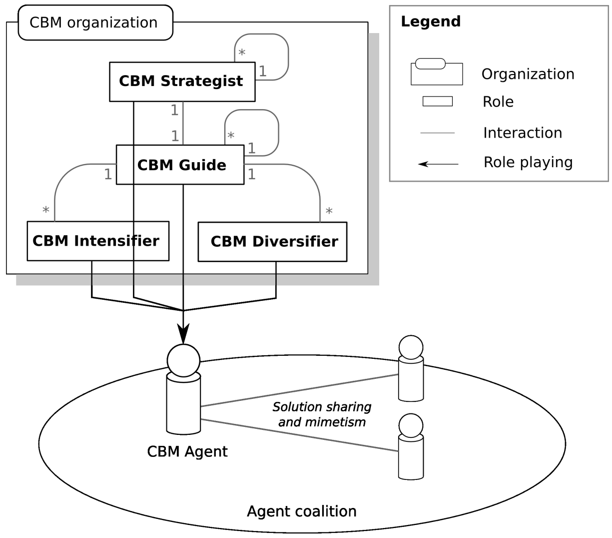

Figure 4: Agentification of CBM organizational model

The behavior of an agent corresponds to a particular schedule of roles. The algorithm 2 presents the behavior of an agent of the coalition. In this algorithm *CBM Intensifier* and *CBM Diversifier* roles correspond to the application of an operator. The *CBM Guide* role is in charge of the choice of the operator and the update of memory. Finally, the *CBM Strategist* role consists of the reinforcement learning and mimetism. This algorithm is close to the AMP scheme described in algorithm 1 with an additional learning step.

Algorithm 2 Role scheduling for a CBM agent

```
Guide role: Initialize the set of solutions and decision process
while stopping criterion is not reached do
    Guide role: Choose operator
    Intensifier or Diversifier role: Apply operator
    Guide role: Update set of solutions
    Strategist role: Learn (reinforcement, mimetism)
end while
```

To complete this description of CBM we detail in the next sections the decision process and learning mechanisms.

4.4 Decision process

The decision process is the main component of the *CBM Guide* role. It allows to select an operator according to the optimization context. To perform the selection of operators we use the mechanism described in [17], which has some similarity with the Holland Classifier Systems, and which is based on a set of rules in form of $\langle \text{condition}, \text{action} \rangle$. Lets C be the set of conditions, O the set of operators. For a condition c_i, a weight $w_{i,j}$ is associated to each operator o_j. The weight $w_{i,j}$ corresponds to the potential of execution of the operator o_j in the condition c_i. The effective choice of an operator is performed by a *roulette wheel selection principle*. Thus, the probability $P(o_j|c_i)$ to apply the operator o_j in the condition c_i is computed using the following formula.

$$P(o_j|c_i) = \frac{w_{i,j}}{\sum_{k=1}^{m} w_{i,k}} \tag{1}$$

With:
$C : (c_i)_{i=1,\ldots,n}$; Set of states
$O : (o_j)_{j=1,\ldots,m}$; Set of operators
$W : (w_{i,j})_{i=1,\ldots,n;j=1,\ldots,m}$; Weight matrix

To select an operator, the optimization context is analyzed to produce an input condition for the decision process. Then an operator is obtained thanks to the *roulette wheel selection principle*. This simple decision process allows to restrain the choice of operators in a given context by setting the corresponding weight value to zero. In addition, the augmentation or diminution of a weight value produce respectively an advantage or a restriction of an operator in a given context. Thus the task of learning mechanisms is to modify the weight values according to the past experiences of the agent.

The determination of the set of conditions is an important step of the design of the decision process. A condition must well characterize the optimization context to take an appropriate decision. Here, the conditions are exclusives and limited to the local minimum properties of the current solution of the agent. Thus, the first condition characterizes a non local minimum solution, and the other conditions correspond to the different combinations of local minimum properties. For instance, if two neighborhood structures $\mathcal{N}_1$ and $\mathcal{N}_2$ are used for the local search, the resulting conditions are:

1. $s_{current}$ is not a local optimum.
2. $s_{current}$ is a local optimum only on $\mathcal{N}_1$.
3. $s_{current}$ is a local optimum only on $\mathcal{N}_2$.
4. $s_{current}$ is a local optimum on $\mathcal{N}_1$ and $\mathcal{N}_2$.

This definition of the set of conditions allows to deal with the frequency and order of application of the operators. Initialization of the weight matrix is made with the parameter α that corresponds to the initial weight value. Thus, after the initialization of the decision process, all possible operators have the same probability to be chosen. Then, during the optimization, the learning mechanisms will adjust the weights according to the experiences of the agent to improve its decision ability.

4.5 Learning mechanisms

The agents use two learning mechanisms to adjust their behaviors, reinforcement learning and mimetism learning, which are related to the *CBM Strategist* role. The learning is performed during the optimization search in order to improve the search strategy of agents. This section describes each of these learning mechanisms.

4.5.1 Reinforcement learning

In [12] the authors define reinforcement learning as the problem faced by an agent that must learn behavior through trial-and-error interactions with a dynamic environment. The two major features of reinforcement learning reported in [18] are trial-and-error search and delayed reward. In our case, several sequences of operators are tried during the optimization thanks to the roulette wheel selection principle. A reinforcement learning is performed if, and only if, the agent find a solution better than its previous best found solution. In this case, the action plan (sequence of operators) which conducted the agent to find this solution is reinforced. Within the decision model previously presented, an experience is a triplet $\langle \text{condition} s_i; \text{operator} o_j; \text{gain} g \rangle$ where the gain is the cost difference of $s_{current}$ obtained by operator application. The reinforcement corresponds to an augmentation of the weight value $w_{i,j}$ related to the experience. This mechanism is intended to favor the behaviors that often find new best solutions.

To perform the reinforcement learning, it is necessary to identify the beneficial experiences and determine a reward. This problem is known as the Credit Assignment Problem. It is difficult to evaluate the efficiency of a given operator immediately after its application since it may depend on the order of application of other operators. Thus, beneficial experiences are identified from the observation of an action sequence performed by the agent. A reinforcement is realized when the current solution fitness is better than the one of the best previously obtained solution of the agent. The experiences from the last diversification operator application to the current state are reinforced.

In order to refine the reinforcement learning, two cases are distinguished, (i) when the agent improves its best found solution, and (ii) when the agent improves the best known solution value it previously obtained during his past interactions with other agents. The reinforcement factors σ_1 and σ_2 are respectively used for the two types of reinforcement. The reinforcement is performed using the formula (2).

$$w_{i,j} = w_{i,j} + \sigma \tag{2}$$

With:
$(c_i; o_j)$; Experience to reinforce
$w_{i,j}$; Weight related to the experience
$\sigma : \{\sigma_1; \sigma_2\}$; Reinforcement factor

4.5.2 Mimetism learning

In the coalition-based metaheuristic, agents perform reinforcement learning individually. The mimetism learning [23] allows a cooperation between agents in order to share the behaviors already enhanced by the reinforcement learning. The mimetism learning works on the assumption that an agent imitates the behaviors of the most efficient agents. At each cycle, the agent examines the fitness value of the best solution found by each other agent of the coalition. When an agent A observes that the agent B has found the best solution value, the agent A imitates the behavior of the agent B.

Lets Wa be the weight matrix of agent A and Wb the weight matrix of agent B, the imitation corresponds to the adoption by agent A of a weight matrix equal to the weighted mean of Wa and Wb. The imitation is computed as follow:

$$Wa = (1 - \rho).Wa + \rho.Wb \quad (3)$$

With:
Wa ; Weight matrix of the imitator agent
Wb ; Weight matrix of the imitated agent
ρ ; Mimetism rate

The combination of reinforcement learning and mimetism learning allows to introduce adaptiveness into the population based search, and then to enhance individual and global behavior. An agent exploits its past experiences in order to improve its capacity to find new best solutions, but it also shares its experiences in order to collectively ensure a better choice of actions in the future. The reinforcement learning allows to improve the local behavior. However, imitation learning lets exploit the search strategies developed by the other agents.

5. CBM FOR SOLVING THE VEHICLE ROUTING PROBLEM

The purpose of this part is to specialize CBM previously presented to solve the Vehicle Routing Problem (VRP). In this section, we present the specialization of CBM to treat the VRP then, some computational results are reported.

5.1 Specializing CBM

The VRP is a well-known problem in the field of transportation and logistics. This problem has been widely studied since 45 years. It consists in finding a set of optimal routes that serve a given set of customers. We use the formulation depicted in [4].

The VRP is defined on a graph $G(V, E)$ where $V = \{v_0, ..., v_n\}$ is a set of vertices and $E = \{(v_i, v_j)/v_i, v_j \in V; i \neq j\}$ represents a set of edges. The vertex v_0 corresponds to the depot while remaining vertices are customers. A quantity q_i of some goods to be delivered by a vehicle and a service time δ_i required by a vehicle to unload the quantity q_i at v_i is associated to each vertex $v_i, i \in \{1, ..., n\}$. A cost or length $c_{i,j}$ is associated to each edge (v_i, v_j). A feasible solution corresponds to a set R of m vehicle routes such that, (i) each route starts and ends at the depot, (ii) each customer is visited exactly once, (iii) the total demand of any route does not exceed the vehicle capacity Q and (iv) the duration of any route does not exceed a bound D. The objective is to minimize the total travel time.

The VRP is NP-hard and can rarely be solved exactly for a number of customers exceeding 100. Several heuristics and metaheuristics have been proposed for the VRP. Surveys on these methods can be found in [6]. CBM seems to be well adapted to solve the VRP since the VRP is a NP-hard combinatorial optimization problem on which a wide range of neighborhood structures can be exploited.

The specialization of CBM for a particular optimization problem requires to define the diversification and intensification operators. The operators used in our approach partially draw from Evolutionary Algorithms. Generation, crossover and mutation operators perform the diversification task. Several standard local search heuristics are used as intensification operators.

Initial solutions are obtained by generation operators. These operators are also used as diversification operators during optimization. Two different operators are used: *greedy insertion algorithm* and *sweep algorithm*. Two crossover operators are used: *route insertion crossover* and *order crossover*. A simple *Remove-and-Reinsert* (RAR) procedure is used as a mutation operator. Four different local search operators are used for the purpose of intensification: *2-opt*, *3-opt*, *1-move* and *1-swap* heuristics. The *2-opt* and *3-opt* heuristics are special case of λ-*opt* heuristics. The *1-move* and *1-swap* heuristics are based on λ-*interchange* mechanisms.

5.2 Computational results

The application of the coalition based metaheuristic to the vehicle routing problem has been tested on the fourteen instances described in Christofides et al. [3]. The CBM has been implemented in Java and tested on a Pentium 4 at 3GHz with 1Gb of memory. The parameter setting of the CBM is given in table 2. The values used for computational experiments are also reported.

The following experiments are performed to assess the improvement of performances resulting from the learning mechanisms proposed, according to different sizes of the coalition, and to evaluate the performances of the approach against some of the powerful heuristics of Operations Research.

Table 2: CBM Parameters

Parameter	Description	Value
α	Initial operator weight value	5
$\sigma_1; \sigma_2$	Rewards for reinforcement learning	1;2
ρ	Mimetism rate	0.4
A	Number of agents in the coalition	15

5.2.1 Performance of reinforcement learning and mimetism

Starting with a referential version of the algorithm with no learning mechanism, we successively introduce the reinforcement learning and the mimetism learning and evaluate the deviation of the average route length to the best known solution values reported in [4].

The CBM has been experimented for different coalition sizes between 1 to 20 agents on the 14 Christofides instances. To make the evaluation fair, the total number of iterations performed by the agents was fixed, and remained constant for all the tests to the value of $10,000$ agent iterations. Thus, in a coalition with A agents, a single agent performs $10,000/A$ iterations. The computation time allowed for each configuration of the algorithm was approximately of 30 seconds, the introduction of the learning mechanisms and the augmentation of the population size having a negligible impact on this value.

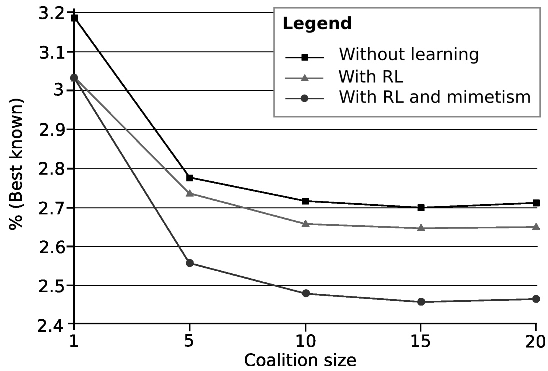

Figure 5: Impact of the learning and the coalition size on the quality of the solution

For each coalition size, CBM is executed 10 times for each of the 14 instances. The average percentage deviations to the best known values are reported in Figure 5. Three different configurations are considered. The first one corresponds to a coalition of agents without Reinforcement Learning (RL) and no mimetism. In this case, the only cooperation mechanism is provided by the standard crossover operators. In the second configuration case, the agents have the capacity to individually learn by reinforcement learning. In the third configuration, both individual and collective learning by mimetism are considered.

It can be observed on the figure that the additional learning capacities improve the quality of the solutions found. In addition, the improvement already carried out by using learning looks to be more pronounced as the population size increases, particularly by using the mimetism learning. Beyond 15 agents, the computational results are slightly deteriorated. This can be explained by the small number of iterations performed by a single agent. The experimentations illustrates the contribution of the cooperation in CBM.

5.2.2 *Evaluation against other metaheuristics*

Here, we evaluate the CBM approach against two powerful metaheuristics presented in the survey of Cordeau et al. [4]. They are: Granular Tabu Search (GTS) [21] and Unified Tabu Search Algorithm (UTSA) [5]. These approaches are selected because they are considered as being ones of the most simple and flexible approaches in the literature. We think that they are the better choice for comparison since CBM also addresses simplicity and flexibility by offering three independent levels of modeling, that are the learning level, the population based metaheuristic level with cooperation, and the problem-dependant heuristic level.

The computational results are presented in Table 3. For each problem, 10 runs are performed and the average and best solution found are considered. The first four columns respectively give the problem name, the type of constraints, the number of customers and the best known solution value taken from [4]. The columns 5-8 respectively report the average deviation of the route length, given in pourcentage, to the best known value, the best found value over the 10 runs, the standard deviation, and the computation time per run in minutes. The other columns report the average route length deviation and the computation time in minutes respectively for the two other approaches. The GTS was evaluated on a Pentium (200 MHz), and the UTSA on a Pentium 4 (2 GHz).

The results indicate that our CBM approach is not yet competitive to the powerful operations research heuristics. Nevertheless, and taking care of the different materials used, with an average deviation of 2.47% and an average computation time of 0.5 minute per run, CBM is not clearly dominated, on both quality and computation time, by the UTSA, which yields 0.56% of deviation in roughly 25 minutes. On the contrary, to be competitive with the GTS, solution quality produced by the CBM, as well as computation time, would have to be improved both by a factor at least 5. However, it is worth to note that, in a first attempt, we use a naive implementation of the operators. We did not use implementation tricks such as candidate lists, "don't look bits", or k-d trees which generally have a great impact on computation times, and then on the overall performances. This point is illustrated here by considering the UTSA which is very slow and the GTS which is very fast. This is because the latter uses such implementation tricks. It is often the case that the most powerful approaches are also the most complicated ones. Such an example is the Active Guided Evolution Strategy (AGES) [13] which is, at the date of writing, the overall winner considering both solution quality and computation time, but which is considered complicated to implement and understand [4].

6. CONCLUSION

In this paper we have proposed AMF, an Agent Metaheuristic Framework that aims at supporting the design and hybridization of metaheuristics. The introduction of an agent-oriented approach allows to deal with flexibility, robustness and modularity in metaheuristics. This framework is based on an organizational model which describes a metaheuristic in terms of roles. Starting from this model, the design process of a metaheuristic consists in refining the AMF organizational model, agentifying the refined model and specializing the multiagent system. The result of this process is a multiagent system that correspond to a particular metaheuristic. In order to illustrate the AMF we have presented an original metaheuristic called CBM using the AMF guidelines. This metaheuristic combines classical metaheuristic approach and DAI concepts. In this paper the efficiency of CBM was illustrated thanks to its application to the Vehicle Routing Problem.

This work is a part of larger effort to provide a whole set of methodological guidelines for the design of metaheuristics to deal with hard combinatorial optimization problems. Further works will deepen the meta-model concepts and associate a methodology to guide the developer during his work of modeling and implementing a multiagent based metaheuristic.

7. REFERENCES

[1] R. Battiti and G. Tecchiolli. The reactive tabu search. *INFORMS Journal on Computing*, 6(2):126–140, 1994.

[2] C. Blum and A. Roli. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Computing Surveys*, 35(3):268–308, 2003.

Table 3: Computational results for the Christofides instances

Instance	Type	Size	Best known	CBM % Average	CBM % Best	CBM St. Dev.	CBM Time	UTSA % Best	UTSA Time	GTS % Best	GTS Time
1	C	50	524.61	**0.00**	**0.00**	0.00	0.06	**0.00**	2.32	**0.00**	0.81
2	C	75	835.26	1.07	0.06	0.77	0.12	**0.00**	14.78	0.40	2.21
3	C	100	826.14	0.38	0.15	0.19	0.31	**0.00**	11.67	0.29	2.39
4	C	150	1 028.42	1.71	0.73	0.44	0.64	0.41	26.66	0.47	4.51
5	C	199	1 291.26	4.05	2.86	0.80	1.10	1.90	57.68	2.09	7.50
11	C	120	1 042.11	14.56	13.86	0.60	0.40	3.01	11.67	0.07	3.18
12	C	100	819.56	1.53	0.19	1.08	0.30	**0.00**	9.02	**0.00**	1.10
6	C, D	50	555.43	0.02	**0.00**	0.07	0.12	**0.00**	3.03	**0.00**	0.86
7	C, D	75	909.68	0.32	**0.00**	0.33	0.25	**0.00**	7.41	1.21	2.75
8	C, D	100	865.94	0.26	**0.00**	0.31	0.47	**0.00**	10.93	0.41	2.90
9	C, D	150	1 162.55	2.55	1.58	1.03	1.25	0.46	51.66	0.91	5.67
10	C, D	199	1 395.85	3.39	2.23	0.85	1.98	1.50	106.28	2.86	9.11
13	C, D	120	1 541.14	4.65	1.56	2.20	0.80	0.53	21.00	0.28	9.34
14	C, D	100	866.37	0.08	**0.00**	0.21	0.39	**0.00**	10.53	**0.00**	1.41
Average C				3.33	2.55	0.55	0.42	0.76	19.11	0.47	3.10
Average C, D				1.61	0.77	0.72	0.75	0.36	30.12	0.81	4.58
Average				2.47	1.66	0.63	0.58	0.56	24.62	0.64	3.84

[3] N. Christofides, A. Mingozzi, and P. Toth. *Combinatorial Optimization*, chapter The vehicle routing problem, pages 315–338. Wiley, 1979.

[4] J. F. Cordeau, M. Gendreau, A. Hertz, G. Laporte, and J. S. Sormany. *Logistics Systems: Design and Optimization*, chapter New Heuristics for the Vehicle Routing Problem, pages 279–297. Springer, 2005.

[5] J. F. Cordeau, G. Laporte, and A. Mercier. A unified tabu search heuristic for vehicle routing problems with time windows. *Journal of the Operational Research Society*, 52:928–936, 2001.

[6] M. Gendreau, G. Laporte, and J.-Y. Potvin. *The Vehicle Routing Problem*, chapter Metaheuristics for the capacitated VRP, pages 129–154. 2002.

[7] F. Glover. Tabu search and adaptive memory programming: advances, applications and challenges. *Advances in metaheuristics, optimization and stochastic modeling technologies*, pages 1–75, 1997.

[8] F. Glover and G. A. Kochenberger. *Handbook of Metaheuristics*. Springer, 2003.

[9] P. Gruer, V. Hilaire, A. Koukam, and K. Cetnarowicz. A formal framework for multi-agent systems analysis and design. *Expert Systems with Applications*, 23(4):349–355, 2002.

[10] V. Hilaire, A. Koukam, P. Gruer, and J.-P. Müller. Formal specification and prototyping of multi-agent systems. *LNAI*, 1972:114–127, 2001.

[11] L. Ingber. Adaptive simulated annealing (asa): Lessons learned. *Journal of Control and Cybernetics*, 25(1):33–54, 1996.

[12] L. P. Kaelbling, M. L. Littman, and A. W. Moore. Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, 4:237–285, 1996.

[13] D. Mester and O. Bräysy. Active guided evolution strategies for large scale vehicle routing problems with time windows. *Computers & Operations Research*, 32:1593–1314, 2005.

[14] S. Meyer-Nieberg and H.-G. Beyer. *Parameter Setting in Evolutionary Algorithm*, chapter Self-Adaptation in Evolutionary Algorithms, pages 47–75. 2007.

[15] M. Milano and A. Roli. Magma: a multiagent architecture for metaheuristics. *IEEE Trans. on Systems, Man, and Cybernetics, Part B*, 34(2):925–941, 2004.

[16] H. V. D. Parunak, S. Brueckner, M. Fleischer, and J. Odell. A design taxonomy of multi-agent interactions. *LNCS*, 2935(4):123–137, 2003.

[17] S. Ropke and D. Pisinger. An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. Technical report, University of Copenhagen, 2005.

[18] R. S. Sutton and A. G. Barto. Reinforcement learning: Introduction. Technical report, Cognitive Science Research Group, 1998.

[19] E. D. Taillard, L. M. Gambardella, M. Gendreau, and J.-Y. Potvin. Adaptive memory programming: A unified view of metaheuristics. *European Journal of Operational Research*, 135:1–16, 2001.

[20] E.-G. Talbi and V. Bachelet. Cosearch: A parallel co-evolutionary metaheuristic. In *Int. workshop on hybrid metaheuritics*, pages 127–140, 2004.

[21] P. Toth and D. Vigo. The granular tabu search and its application to the vehicle routing problem. *INFORMS Journal on Computing*, 15:333–348, 2003.

[22] S. Voss. *Local Search for Planning and Scheduling*, volume 2148 of *LNCS*, chapter Meta-heuristics: The State of the Art, pages 1–23. 2001.

[23] T. Yamaguchi, Y. Tanaka, and M. Yachida. Speed up reinforcement learning between two agents with adaptive mimetism. In *IEEE Int. Conf. on Intelligent Robots and Systems*, volume 2, pages 594–600, 1997.