A Cooperative and Self-adaptive Metaheuristic for the Facility Location Problem

David Meignan
Laboratoire Systèmes et Transports
Université de Technologie de Belfort-Montbéliard
Belfort, France
david.meignan@utbm.fr

Jean-Charles Créput
Laboratoire Systèmes et Transports
Université de Technologie de Belfort-Montbéliard
Belfort, France
jean-charles.creput@utbm.fr

Abderrafiaa Koukam
Laboratoire Systèmes et Transports
Université de Technologie de Belfort-Montbéliard
Belfort, France
abder.koukam@utbm.fr

ABSTRACT

This paper presents a coalition-based metaheuristic (CBM) to solve the uncapacitated facility location problem. CBM is a population-based metaheuristic where individuals encapsulate a single solution and are considered as agents. In comparison to classical evolutionary algorithms, these agents have additional capacities of decision, learning and cooperation. Our approach is also a case study to present how concepts from multiagent systems' domain may contribute to the design of new metaheuristics. The tackled problem is a well-known combinatorial optimization problem, namely the uncapacitated facility location problem, that consists in determining the sites in which some facilities must be set up to satisfy the requirements of a client set at minimum cost. A computational experiment is conducted to test the performance of learning mechanisms and to compare our approach with several existing metaheuristics. The results showed that CBM is competitive with powerful heuristics approaches and presents several advantages in terms of flexibility and modularity.

Categories and Subject Descriptors

I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search—*Heuristic methods*

General Terms

Algorithms

Keywords

combinatorial optimization, metaheuristic, multiagent system, facility location problem

GECCO'09, July 8–12, 2009, Montréal, Québec, Canada.

1. INTRODUCTION

Research in metaheuristics usually focuses on performance facing large or dynamic problem instances. However, recent studies tend to put the emphasis on simplicity, flexibility and modularity of metaheuristics. These features which constitute important criteria for an effective metaheuristics use, have been put forward in several articles and surveys [26, 2].

In [4], the authors defined the simplicity and flexibility criteria in these terms: "Simplicity relates to ease of understanding and coding of an algorithm" and "Flexibility measures the capacity of adapting an algorithm to effectively deal with additional constraints". In addition, robustness can be viewed as the ability to solve different instances of a same problem while maintaining computational performance. Finally, modularity is the capacity of an algorithm to be reused, hybridized or paralleled.

Distributed artificial intelligence 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 can exploit the social metaphor and self-organization paradigm. Thus, multiagent concepts are widely used in metaheuristics, particularly for population-based, hybrid and distributed metaheuristics. For instance, the concept of agent is explicitly used in Co-search metaheuristic [25] or MAGMA's metaheuristics architecture [18]. The advantages of using multiagent approach for metaheuristics may be justified by the distribution and robustness inherent to multiagent systems and the need of flexibility and modularity.

Our objective in this paper is to present a Coalition-Based Metaheuristic (CBM) which combines evolutionary algorithms approach and distributed artificial intelligence concepts to solve the Uncapacitated Facility Location Problem (UFLP). We detail particularly how the metaheuristic take advantages of learning and cooperation mechanisms. In addition, we present the usage of an Agent Metaheuristic Framework (AMF).

CBM adopts a decentralized approach close to fine-grained evolutionary algorithms [1]. Indeed, individuals which encapsulate a single solution are considered as agents that are organized in a group called a coalition. In comparison to simple evolutionary algorithms, these agents have additional capacities of decision, learning and cooperation. The coalition structure is intended to facilitate the distribution since the removal and addition of any agent does not perturb the global functioning of the system. In addition, the learning and cooperation capacities of agents favor the adaptation to various problem instances. Finally, the use of a generic model of metaheuristics drawn from AMF (Agent Metaheuristic Framework) [17] assists the modularity and reusability of CBM.

This paper is organized as follows. Section 2 introduces the Agent Metaheuristic Framework. In section 3, we present the Coalition-Based Metaheuristic. Then, section 4 is devoted to the application of the metaheuristic to a facility location problem. The last section gives some conclusions and perspectives.

2. AN AGENT METAHEURISTIC FRAMEWORK

The Coalition-Based Metaheuristic which is presented in the next section is based on the Agent Metaheuristic Framework (AMF) [17]. This framework aims at analyzing existing algorithms, and facilitating the design of hybrid or new metaheuristics. It proposes an organizational model of metaheuristics that can be used to describe either population-based metaheuristics or trajectory methods. In fact, a metaheuristic is viewed as an organization composed of a set of roles which interact in order to find an optimal solution.

The organizational model of AMF uses the concepts of role, interaction and organization [11]. A role is an abstraction of a behavior or a status in an organization. An interaction links two roles in such 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 associated to a global task or goal to satisfy. From these concepts, a metaheuristic is defined as an organization. The goal of this organization is to efficiently explore the search space in order to find near-optimal 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 the search experiences. Four roles stems from this definition: Intensifier, Diversifier, Guide and Strategist. The resulting metaheuristic organizational model is described in figure 1. The definitions of the four roles composing the metaheuristic model are given below.

The Intensifier and Diversifier roles respectively represent the intensification and diversification procedures or tendencies. Thus, the goal of the Intensifier role is to concentrate the search in promising areas of the search space. On the contrary, the goal of Diversifier role is to move the search to unexplored area. A comprehensive study of the concepts of intensification and diversification in metaheuristics can be found in [2].

In a metaheuristic, these two roles can refer to a single procedure or two distinct ones. For instance, in the iterated local search metaheuristic [16], intensification is performed by a local descent procedure and diversification corresponds to a perturbation procedure. In the ant colony optimization metaheuristic, intensification and diversification tendencies can be identified [7] but they are combined in the decision process of the ants. Thanks to the concept of role, the AMF organizational model manages these two cases.

The goal of the Guide role is to balance diversification and intensification tendencies, and to coordinate diversifier and intensifier roles. The main element of the Guide role is the memory that stores and provides information for the intensification and diversification. The term "memory" draws from the Adaptive Memory Programming (AMP) scheme [24]. Memory 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.

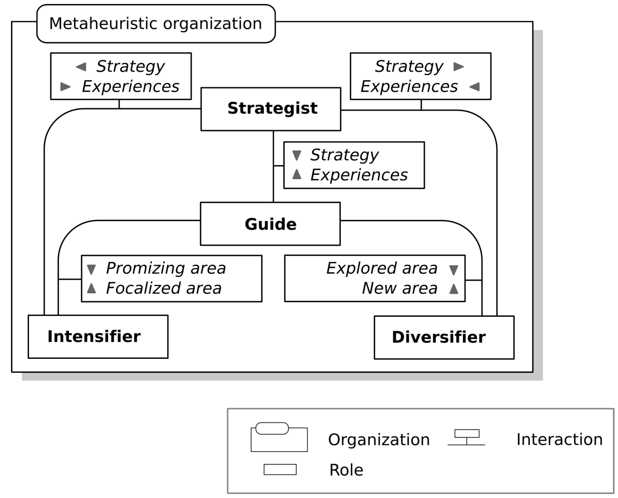

Figure 1: AMF organizational model of metaheuristics

On the contrary to the first three roles, the Strategist role is not implemented in all metaheuristics. This role corresponds to the adaptation or self-adaptation mechanisms in metaheuristics. The goal of the Strategist role is to improve the performance of the search process and possibly reduce the parameter setting. The concept of adaptation in AMF is close to the definition given by Hinterding et al. for evolutionary computation [12]. In AMF, adaptation is characterized by the modification or adjustment of the search strategy resulting from the observation of experiences. Thus, adaptation mechanisms use some kind of feedback to determine the nature or amplitude of the change.

The organizational model of AMF can be considered as a pattern for metaheuristics and several metaheuristics can be analyzed from the roles introduced in the model. This approach assists the modularity and encourages the design of distributed and adaptive metaheuristics. In the next section, this model is used to define the agent's architecture of the coalition metaheuristic.

3. A COALITION-BASED METAHEURISTIC

In the previous part, an organizational model for the design and hybridization of metaheuristics has been described. In this section, the AMF model is used to define agent's architecture of a Coalition-Based Metaheuristic (CBM). This metaheuristic is based on the metaphor of a coalition where agents cooperate to treat an optimization problem. We first describe the main features of CBM, then, we put the emphasis on the decision process and learning mechanisms related to the agents.

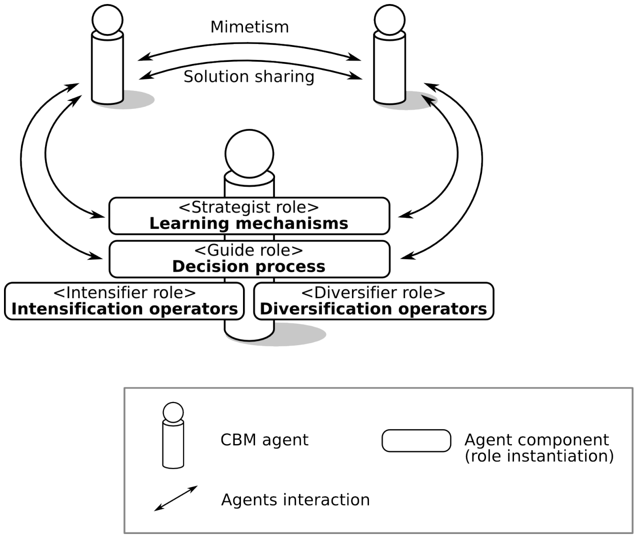

Figure 2: Agent architecture in CBM

3.1 CBM principles and agents architecture

CBM is a metaheuristic based on the metaphor of the coalition. In the multi-agent domain, a coalition is a flat structure where agents have the same capacities and cooperate by the mean of direct interactions. Agent's cooperation has to contribute to the realization of a common task [19]. The coalition structure is intended to support robustness and facilitate the distribution since the control is decentralized, the communication between agents are asynchronous, and consequently, the removal or addition of any agent do not perturb the global functioning of the system.

In CBM, the coalition is composed of several agents which have the capacity to treat the optimization problem but cooperate to coordinate the search and improve their search abilities. To perform the search, an agent manages three solutions as in particle swarm optimization [14]: a current solution, a best found solution and a best known solution. An agent uses several operators which are applied on its current solution (see figure 2). These operators constitute the low level of the agents and correspond to the Intensifier and Diversifier roles in the AMF model. Intensification operators refer to improvement process such as local search procedure, and diversification operators correspond to generation, mutation or crossover procedures.

The schedule of the operators is determined by a decision process. It corresponds to the Guide role in the AMF model. The objectives of this component consist in selecting the most appropriate operators and coordinating intensification and diversification procedures. The selection of operators, detailed in the next section, is based on heuristics rules and follows the hyper-heuristics approach [3]. The search behavior of an agent is adapted during the optimization process by learning mechanisms. These mechanisms modify the rules of the decision process according to the experiences of the agents.

The agents have two cooperation means. In one hand, an agent can share its best found solutions with the rest of the coalition. These solutions are exploited by other agents for crossover operators or to compare agents' search abilities. In another hand, an agent can share its internal decision rules in order to enable mimetism of behavior. This second cooperation mechanism is intended to favor the search behaviors that often found new best solutions.

By using the AMF model, the architecture of the agents in CBM is modular and reusable. Indeed, each components presented in figure 2 can be separately specified. In addition, thanks to the hyper-heuristic approach, the decision process and learning mechanisms are problem independent.

3.2 Decision process

The decision process allows the selection of operators according to the optimization context. To perform the selection of operators we use a mechanism close to the ALNS (Adaptive Large Neighborhood Search) approach [21]. It is based on a set of rules in form of $\langle \text{condition}, \text{action} \rangle$, where the actions correspond to the intensification and diversification operators.

Let 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 selection 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

This simple decision process enables to restrain the choice of operators in a given state 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 state. Thus the task of learning mechanisms is to modify the weight values according to the past experiences of the agent.

The set of states has been chosen to obtain an alternation between intensification and diversification operators and to manage the order and frequency of operators. Thus, each state corresponds to the previous application of operators. The first state corresponds to the previous application of one of the diversification operator. The next states are associated to the previous application of each intensification operators. The last state is activated only when all intensification operator have been applied without modifying the current solution. Since all intensification operators correspond to local descent procedures, this state characterizes a local optimum on all used neighborhood structures.

The weight values in matrix W is initialized with parameter α and several values are set to 0. This initialization determines a cycle in the application of intensification and diversification operators. The cycle alternates the application of several intensification operators until the last state is reach, then the application of a diversification operator. Even if the operators' choice is restricted by this initialization, it is still possible to modify the order and frequency of operators' selection in the cycle. This modification is performed by learning mechanisms.

3.3 Learning mechanisms

The agents jointly use two learning mechanisms to adjust their behaviors, reinforcement learning and mimetism learning. The learning is performed during the optimization search in order to improve the search strategy of agents. This section briefly describes each of these two learning mechanisms.

3.3.1 Reinforcement learning

In [13], 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 [23] are trial-and-error search and delayed reward.

In CBM, the problem of selecting the most appropriate operators is viewed as a reinforcement learning problem. During the optimization process, an agent tries several sequences of operators and it must learn from these experiences. Within the decision model previously presented, an experience is defined as a triplet $\langle$condition c_i; operator o_j; gain $g\rangle$ where the gain is the fitness difference obtained by the operator application. 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 a diversification/intensification cycle. A reinforcement is realized at the end of a cycle and when a new best found solution has been reached. In this case, the experiences with a non null gain, from the last diversification operator application to the current state are reinforced.

Figure 3 presents a typical case where reinforcement learning is applied. The cost of the best found solution and the current solution of an agent are plotted. After the application of a diversification operator (o_3) and of several intensification operators (o_1, o_2), the agent improves the cost of its best found solution. Then, a reinforcement is applied on the experiences $\langle c_4; o_3; 10\rangle$, $\langle c_1; o_1; -8\rangle$ and $\langle c_2; o_2; -4\rangle$. Thus, the weight $w_{4,3}$, $w_{1,1}$ and $w_{2,2}$ are augmented to favor the selection of the operators in the same conditions.

In order to refine 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 in the coalition. 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:
$\langle c_i; o_j; g\rangle$; Experience to reinforce
$w_{i,j}$; Weight related to the experience
$\sigma : \{\sigma_1; \sigma_2\}$; Reinforcement factor

3.3.2 Mimetism

In CBM, agents perform reinforcement learning individually. The mimetism learning [27] allows cooperation between agents in order to share the behaviors already enhanced by

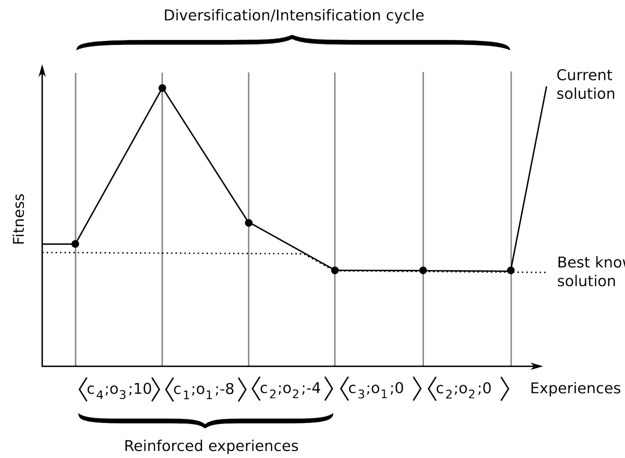

Figure 3: Case of reinforcement learning

the reinforcement learning. The mimetism learning works on the assumption that an agent tends to behave as 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. Let W_a be the weight matrix of agent A and W_b 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 W_a and W_b. The imitation is computed as follow:

$$Wa = (1 - \rho).Wa + \rho.Wb \tag{3}$$

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 enables to improve the local behavior. However, mimetism learning lets exploit the search strategies developed by the other agents.

4. CBM FOR THE UNCAPACITATED FACILITY LOCATION PROBLEM

In this section we present the specialization of CBM to solve the Uncapacitated Facility Location Problem (UFLP). Computational results are reported in th next section to confirm the improvement of performances resulting from the learning mechanisms. Our approach is then compared with several existing metaheuristics.

4.1 The uncapacitated facility location problem

The UFLP is a widely studied $\mathcal{NP}$-Hard problem in combinatorial optimization [5]. It can be described as the problem of choosing the sites in which facilities must be set up to satisfy the demands of a given set of clients at minimum costs. This location problem is uncapacitated since the number of clients allocated to a facility is not limited. The UFLP can be formally defined in the following way.

Let $I = \{i_1, ..., i_m\}$ be the set of candidate sites in which facilities can be located, $J = \{j_1, ..., j_n\}$ the set of clients. For each site $i \in I$ a setup cost f_i is defined. The transportation cost between a site $i \in I$ and client $j \in J$ is given by c_{ij}. The UFLP consists in determining a set S of sites to open so as to minimize the total cost $C(S)$.

$$\text{Minimize } C(S), \emptyset \subset S \subseteq I \tag{4}$$

with,

$$C(S) = \sum_{i \in S} f_i + \sum_{j \in J} min\{c_{ij} | i \in S\} \tag{5}$$

4.2 CBM specialization for UFLP

The specialization of CBM for a particular optimization problem necessitates the definition of 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 local descent heuristics are used as intensification operators. To solve the UFLP, five diversification operators and two intensification procedures are used by the agents.

The two intensification operators are based on two different neighborhood structures, *1-switch* and *2-swap*. The *1-switch* neighborhood of a solution is obtained by opening or closing one site in the initial solution. A move in the *2-swap* neighborhood of a solution consists in modifying the status of two sites; one site is opened while the second is closed. These two neighborhood structures are used in a local descent procedure. This local search procedure consists in performing a sequence of moves towards a local optimum solution. It uses a *first improvement policy* to select a solution in a randomly ordered neighborhood [10].

The set of diversification operators is composed of two generation heuristics, two mutations procedures and a crossover operator.

The first generation operator generates a solution by opening a randomly number of sites, then closing the sites which have none allocated clients. The second generation operator is a *greedy construction* procedure which start with a solution without opened site, then progressively open the sites while the solution cost is improved.

The two mutation operators correspond to a perturbation procedure based on the *1-switch* neighborhood structure. These two operators differ in the perturbation amplitude (number of moves applied).

The last diversification operator is a *uniform crossover*. To obtain a new solution, this procedure randomly chooses the status of sites between the two initial solutions.

5. COMPUTATIONAL RESULTS

The application of the coalition based metaheuristic to the UFLP has been tested on the ninety instances of the *Körkel-Ghosh* benchmark. The benchmark has been proposed in [8] and follows the generation principle described in [15]. It is composed of large instances from 250×250 to 750×750 (number of sites × number of clients) divided into several classes (symmetric/asymmetric transportation cost matrix, low/medium/high setup costs).

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 5. The following experiments are performed to confirm the improvement of performances resulting from the learning mechanisms and to compare our approach with several existing metaheuristics.

Parameter	Description	Value
α	Initial operator weight value	1.0
$\sigma_1; \sigma_2$	Reinforcement factors	0.5; 0.75
ρ	Mimetism rate	0.3
A	Number of agents in the coalition	15

Table 1: Parameter setting of CBM for UFLP

5.1 Performance of reinforcement learning and mimetism

To evaluate reinforcement learning and mimetism in CBM, three configurations which differ on activated learning mechanisms are considered. The first one corresponds to a coalition of agents without reinforcement learning (RL) and no mimetism. In the second configuration, the agents have the capacity to individually learn by reinforcement learning. In the last configuration, both individual and collective learning by mimetism are considered.

These three configurations of CBM have been experimented on the five first instances of the *Körkel-Ghosh* benchmark. For each configuration and instance, CBM was run 25 times with 20 seconds of computational time. Figure 4 reports the average deviation in % to the best known solutions values reported in [22].

It can be observed on the figure that the addition of reinforcement learning and the addition of mimetism capacities improve the quality of the solutions found. Note for the second instance that the three configurations of CBM always found the best known solution value. This experimentation illustrates the positive impact of learning mechanisms on the agents' behavior. During the optimization process, the agents modify the rules of their decision processes and improve their capacity to choose the most appropriate operators. It results an improvement of the final solutions values.

5.2 Comparative results

CBM have been compared with four powerful heuristics: two tabu search approaches [8, 22], an original trajectory method called CLM (Complete Local Search with Memory) [9], and a hybrid population-based metaheuristic [20].

The tabu search approach of Ghosh [8] uses a neighborhood which corresponds to a combination of the *1-switch* and *2-swap* neighborhood structures. The tabu list is based on a recency-based memory that discourages the modification of sites that were involved in recent moves. In addition, a frequency-based memory keeps a record of the number of times a site has participated in a move during the history of the search. A penalty function based on this memory penalizes the use of sites that have a high frequency value.

The tabu search developed by Sun [22] differs from the Ghosh one by using the *1-switch* neighborhood structure and adding a long term memory in addition to recency-based and

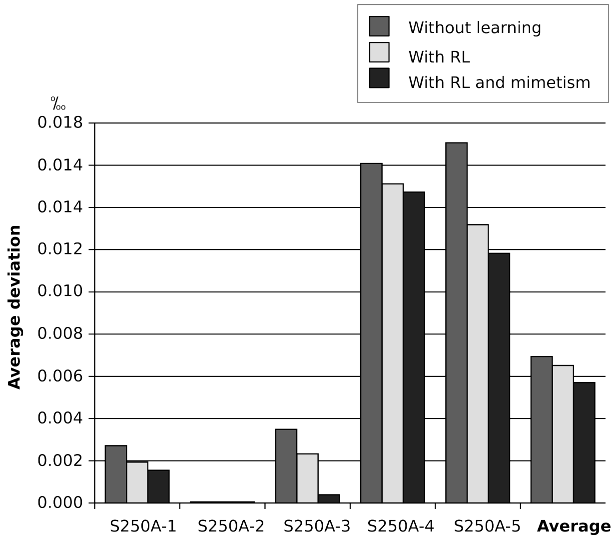

Figure 4: Performances of CBM with an without learning mechanisms on Körkel-Ghosh S250A instances

frequency-based memory. This long term memory is used to achieve diversification steps to lead the search in new regions in the solution space.

CLM is a graph search based approach proposed by Ghosh and Sierksma in [9]. For the UFLP, the graph is defined from the *1-switch* and *2-swap* neighborhood structures. The search is performed by manipulating two sets called LIVE and DEAD. LIVE contains solutions that are available for the exploration and DEAD contains solutions that have been already considered by heuristic. The global procedure consists in iteratively explore the neighborhood of solutions in LIVE. In addition, this search is improved by a local descent procedure applied in particular conditions.

The Hybrid algorithm, proposed by Resende and Werneck in [20], combines some elements of scatter search and evolutionary algorithms. The method works in two phases. The first one is a multistart procedure. Each iteration aims at, building a randomized solution, then applying a local search, and finally combining the resulting solution with previously ones through a path-relinking procedure. The second phase combines the pool of elite solutions also with a path-relinking procedure. In the first phase, the selection of the pool of elite solutions is based on a bi-criteria rule. This selection rule favors both solution fitness and distance between solutions.

The evaluations of these four heuristics and CBM have been made on the *Körkel-Ghosh* benchmark. To compare computational time coming from different computers, we normalize CPU times with the factors presented in table 2. These factors derive from the performance measure in Mflop/s reported in [6].

The computational results are presented in table 3. Each line represents a group of five instances, thus each value corresponds to an average on these five instances. The first four columns respectively give the problem size, the type of the transportation cost matrix c_{ij} (Symmetric/Asymmetric), the type of setup cost (A for low, B for medium and C for high) and the best known average value reported in [22]. The

Computer	Estimated performance (Mflop/s)	Normalization factor
Pentium IV 3GHz	1571	1
Celeron 650MHz	395	1/3,98
SGI Challenge 196MHz MIPS R10000	130	1/12,08
Sun Enterprise 3000	110	1/14,28

Table 2: Estimated computers performances

next two columns correspond to the deviation in % to the best known values and the computational times in seconds for CBM. For each instance, 10 runs are performed and the average values are considered. The other columns report the deviation to the best known values and the normalized computational times respectively for CLM [9], Tabu Search (TS) of Ghosh and Sierksma [9], the Hybrid algorithm of Resende and Werneck [20] and the Tabu Search with Long-Term Memory (TS-LTM) of Sun [22]. Note that all deviation values of TS-LTM are null because they are used as best known values.

The results indicate that our CBM approach gives better solution values than TS and CLM on each group of instances with slightly smaller computational times. Considering solution quality, an average deviation of $0,009\%$ is promising, but CBM is dominated by Hybrid and TS-LTM approach. CBM is not yet competitive to the computational times of Hybrid and TS-LTM. This can be improved for instance by a better implementation of the structures used to evaluate the solutions costs, or by reducing the neighborhood size used by intensification operators.

Some additional criteria such as flexibility and modularity have to be considered to evaluate the different approaches since CBM addresses these issues. Flexibility can be defined as the capacity of adapting an algorithm to effectively deal with additional constraints. By extension, an algorithm which is highly problem dependent cannot be considered as flexible. On this criterion, the two tabu search approaches seem to be less flexible than CBM, CLM and Hybrid. Indeed, the implementations of the tabu search by Ghosh [8] and Sun [22] exploit the problem structure to define recency-based, frequency-based and long-term memory.

The modularity is the capacity of an algorithm to be reused, hybridized or paralleled. Considering this criterion, CBM has several advantages. First of all, new intensification and diversification operators can be easily introduced without modifying the agents' architecture. These operators are automatically managed thanks to the decision process and learning mechanisms. Then, by using the AMF model, others decision or learning procedures can be considering. Finally, the decentralization in CBM and the asynchronous nature of agents' interactions favor the parallelization.

6. CONCLUSION

In this paper we have introduced CBM, a new multiagent metaheuristic to solve the uncapacitated facility location problem. This metaheuristic have been designed using the Agent Metaheuristic Framework (AMF) that provides a generic model of metaheuristics.

CBM adopts a decentralized approach close to fine-grained

Table 3: Computational results on *Körkel-Ghosh* benchmark

Instance				CBM		CLM		TS		Hybrid		TS-LTM	
Size	Type		Best known	Dev. (%)	Time (s)[a]	Dev. (%)	Time (s)[b]	Dev. (%)	Time (s)[b]	Dev. (%)	Time (s)[c]	Dev. (%)	Time (s)[d]
250	Sym.	A	257805,0	0,006%	3,7	0,035%	4,6	0,011%	4,6	0,001%	0,4	0,000%	0,2
		B	276035,2	0,003%	3,2	0,115%	1,6	0,054%	1,6	0,000%	0,6	0,000%	0,4
		C	333671,6	0,000%	4,2	0,000%	4,4	0,044%	4,4	0,000%	0,7	0,000%	0,7
	Asym.	A	257917,8	0,008%	4,7	0,045%	4,5	0,023%	4,5	0,002%	0,4	0,000%	0,2
		B	276053,2	0,004%	3,5	0,047%	1,6	0,150%	1,6	0,000%	0,6	0,000%	0,4
		C	332897,2	0,005%	4,3	0,048%	6,2	0,104%	6,2	0,000%	0,6	0,000%	0,6
500	Sym.	A	511180,4	0,016%	32,0	0,060%	53,6	0,040%	53,6	0,003%	3,4	0,000%	1,1
		B	537912,0	0,017%	22,8	0,144%	17,9	0,106%	17,9	0,001%	3,7	0,000%	2,2
		C	621059,2	0,001%	29,5	0,018%	36,8	0,008%	36,8	0,000%	3,9	0,000%	5,0
	Asym.	A	511140,0	0,007%	36,8	0,050%	52,0	0,022%	52,0	0,002%	2,8	0,000%	1,0
		B	537847,6	0,013%	30,8	0,107%	19,9	0,055%	19,9	0,005%	3,5	0,000%	2,4
		C	621463,8	0,012%	40,3	0,085%	33,8	0,067%	33,8	0,001%	4,5	0,000%	5,0
750	Sym.	A	763693,4	0,011%	105,9	0,037%	207,1	0,018%	207,1	0,002%	7,6	0,000%	2,8
		B	796571,8	0,019%	72,3	0,076%	102,9	0,044%	102,9	0,004%	8,9	0,000%	6,5
		C	900158,6	0,006%	95,5	0,070%	87,3	0,111%	87,3	0,004%	9,8	0,000%	16,1
	Asym.	A	763717,0	0,011%	109,4	0,040%	211,9	0,016%	211,9	0,003%	8,1	0,000%	2,8
		B	796374,4	0,012%	74,8	0,061%	99,5	0,061%	99,5	0,001%	9,5	0,000%	6,7
		C	900193,2	0,004%	94,7	0,017%	125,6	0,036%	125,6	0,000%	10,4	0,000%	16,6
Average				0,009%	42,7	0,059%	59,5	0,054%	59,5	0,002%	4,4	0,000%	3,9

[a] Average time per run on a Pentium 4 at 3GHz.
[b] Time per run on a Celeron 650MHz, normalized to Pentium 4 at 3GHz.
[c] Time per run on a SGI Challenge R10000, normalized to Pentium 4 at 3GHz.
[d] Time per run on a Sun Enterprise 3000, normalized to Pentium 4 at 3GHz.

evolutionary algorithms. Individuals which encapsulate a single solution are considered as agents that are organized in a group called a coalition. In comparison to simple evolutionary algorithms, these agents have additional capacities of decision, learning and cooperation. To perform the search, an agent uses several operators which are scheduled by an adaptive decision process. This decision process is based on heuristics rules and follows the hyper-heuristic approach in the sense that it is problem independent. In addition, the decision rules of the agents are adapted during the optimization process by reinforcement learning and mimetism.

CBM exploits several aspects of multiagent systems. The CBM agent architecture is based on the Agent Metaheuristic Framework (AMF) which encourages modularity and reusability. Then, the coalition structure is intended to support robustness and facilitate the distribution, since control is decentralized and the agents' interactions are asynchronous. Finally, cooperation and learning mechanisms contribute to the effectiveness of the optimization.

The metaheuristic has been applied to the uncapacitated facility location problem. Experiments were performed to confirm the efficiency of learning mechanisms. Our approach was also compared with several existing metaheuristics. Computational results indicate that our approach is competitive to the best heuristics in term of solution quality. In addition, CBM has several advantages considering modularity and flexibility criteria. In further works, computational times should be improved by a better implementation of the problem-dependent operators. These works may also consider the comparison of the decision and learning mechanisms in CBM with hyper-heuristics ones.

7. REFERENCES

[1] E. Alba and M. Tomassini. Parallelism and evolutionary algorithms. *IEEE Transaction on Evolutionary Computation*, 6(5):443–462, 2002.

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

[3] E. Burke, E. Hart, G. Kendall, J. Newall, E. Hart, P. Ross, and S. Schulenburg. *Handbook of Meta-Heuristics*, chapter Hyper-Heuristics: An Emerging Direction in Modern Search Technology, pages 457–474. Kluwer Academic, 2003.

[4] J.-F. Cordeau, M. Gendreau, A. Hertz, G. Laporte, and J.-S. Sormany. New heuristics for the vehicle routing problem. In *Les Cahiers du GERAD*. 2004.

[5] G. Cornuéjols, G. L. Nemhauser, and L. A. Wolsey. *Discrete Location Theory*, chapter The uncapacitated facility location problem, pages 199–171. John Wiley and Sons, New York, 1990.

[6] J. J. Dongarra. Performance of various computers using standard linear equations software. Technical Report CS-89-85, Computer Science Department, University of Tennessee and Computer Science and Mathematics Division, Oak Ridge National Laboratory, 2006.

[7] M. Dorigo and T. Stützle. The ant colony optimization metaheuristic: Algorithms, applications and advances. Technical Report IRIDIA-2000-32, IRIDIA, 2000.

[8] D. Ghosh. Neighborhood search heuristics for the uncapacitated facility location problem. *European Journal of Operational Research*, 150:150–162, 2003.

[9] D. Ghosh and G. Sierksma. Complete local search with memory. *Journal of Heuristics*, 8(6):571–584, 2002.

[10] P. Hansen and N. Mladenović. *Handbook of Metaheuristics*, chapter Variable Neighborhood Search, pages 145–184. Kluwer Academic, 2003.

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

[12] R. Hinterding, Z. Michalewicz, and A. E. Eiben. Adaptation in evolutionary computation: a survey. In *IEEE International Conference on Evolutionary Computation*, pages 65–69, 1997.

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

[14] J. Kennedy and R. C. Eberhart. Particle swarm optimization. In *IEEE International Conference on Neural Networks*, pages 1942–1948, 1995.

[15] M. Körkel. On the exact solution of large-scale simple plant location problems. *European Journal of Operational Research*, 39(2):157–173, 1989.

[16] H. R. Lourenço, O. C. Martin, and T. Stützle. *Handbook of Metaheuristics*, chapter Iterated Local Search, pages 321–353. Kluwer Academic, 2003.

[17] D. Meignan, J.-C. Créput, and A. Koukam. An organizational view of metaheuristics. In N. R. Jennings, A. Rogers, A. Petcu, and S. D. Ramchurn, editors, *First International Workshop on Optimisation in Multi-Agent Systems, AAMAS'08*, pages 77–85, 2008.

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

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

[20] M. Resende and R. Werneck. A hybrid multistart heuristic for the uncapacitated facility location problem. *European Journal of Operational Research*, 174(1):54–68, 2006.

[21] S. Ropke and D. Pisinger. An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transportation Science*, 40:421–438, 2006.

[22] M. Sun. Solving the uncapacitated facility location problem using tabu search. *Computers & Operations Research*, 33(9):2563–2589, 2006.

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

[24] 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.

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

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

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