Max Min Ant System and Capacitated *p*-Medians: Extensions and Improved Solutions

Fabrício Olivetti de França and Fernando J. Von Zuben
LBiC/DCA/FEEC
State University of Campinas (Unicamp)
PO Box 6101, 13083-852 – Campinas/SP, Brazil

Leandro Nunes de Castro
Research and Graduate Program on Informatics
Catholic University of Santos (UniSantos)
R. Dr. Carvalho de Mendonça, 144
11070-906, Santos/SP, Brazil

Keywords: ant colony optimization, capacitated p-medians, clustering problems.

Received: September 24, 2004

This work introduces a modified MAX MIN Ant System (MMAS) designed to solve the Capacitated p-*Medians Problem (CPMP). It presents the most relevant steps towards the implementation of an MMAS to solve the CPMP, including some improvements on the original MMAS algorithm, such as the use of a density model in the information heuristics and a local search adapted from the uncapacitated* p-*medians problem. Extensions of a recently proposed updating rule for the pheromone level, aiming at improving the MMAS ability to deal with large-scale instances, are also presented and discussed. Some simulations are performed using instances available from the literature, and well-known heuristics are employed for benchmarking.*

Povzetek: Predstavljene so izboljšave algoritma MAX MIN na osnovi kolonij mravelj za reševanje problema CPMP.

1 Introduction

The capacitated *p*-medians problem (CPMP), also known as capacitated clustering problem, is a combinatorial programming task that can be described as follows: given a graph with *n* vertices (clients), find *p* centers (medians) and assign the other vertices to them minimizing the total distance covered, limited to a capacity restriction. This problem is a special case of the “capacitated plant location problem with single source constraints” and many other combinatorial problems as pointed in Osman and Christofides [1]. As such, the CPMP was proved to be NP-complete in Garey and Johnson [2]. Its practical use varies from industrial and commercial planning to every clustering related problem, like data mining, pattern recognition, vehicle routing and many others.

Ant Systems (AS) were first proposed in Dorigo [3] as an attempt to use the ant foraging behavior as a source of inspiration for the development of new search and optimization techniques. By using the pheromone trail as a reinforcement signal for the choice of which path to follow, ants tend to find “minimal” routes from the nest to the food source. The system is based on the fact that ants, while foraging, deposit a chemical substance, known as pheromone, on the path they use to go from the food source to the nest. The standard system was later extended in Dorigo and Di Caro [4], giving rise to the so-called Max Min Ant System (MMAS). The main purpose of the max-min version is to improve the search capability of the standard algorithm by combining exploitation with exploration of the search space, and by imposing bounds to the pheromone level, thus helping to avoid stagnation.

This paper is an extension of the work initiated in de França et al. [5], with additional contributions: a thorough analysis and explanation of the proposed operators, and a broader set of experiments. Essentially, the innovative aspects of the approach are twofold: (*i*) adaptation of the MMAS algorithm to deal with a problem not previously conceived by means of an ant-based formalism; and (*ii*) proposition of several modifications to the MMAS algorithm so as to improve its performance when dealing with large instances of combinatorial optimization problems. In practical terms, the ant system will incorporate a local search procedure for the CPMP, a new updating rule for the pheromone level, and a stagnation control mechanism.

The paper is organized as follows. Section 2 provides a mathematical formulation of the CPMP problem and the General Assignment Problem (GAP) that results when the medians are already specified. In Section 3, the basic Ant System algorithm together with its Max Min version, MMAS, are reviewed. Section 4 emphasizes the contributions of this work. It describes the proposed enhancements of MMAS, leading to the improved MMAS, called here IMMAS, and how to apply ant-based algorithms to the capacitated *p*-medians problem. The proposed algorithm is evaluated in Section 5, and its performance is compared with that of other works from the literature. The paper is concluded in Section 6 with a discussion about the formal and methodological contributions and a description of several avenues for further investigation.

2 Mathematical Formulation of the Capacitated p-Medians Problem

This section provides a mathematical formulation of the capacitated p-medians problem as a constrained optimization problem: the total distance from the medians to the clients has to be minimized, constrained by the demands of clients and capacities of medians.

On a complete graph, given n nodes with predefined capacities and demands, the goal is to choose p nodes ($p < n$) as capacitated medians and to attribute each one of the remaining ($n - p$) nodes, denoted clients, to one of the chosen medians, so that the capacity of each median is not violated by the cumulated demand, and the sum of the distances from each client to the corresponding median is minimal. Every node is a candidate to become a median, and the solution will consider demand and capacity of medians, and only demand of clients.

Defining an $n \times n$ matrix $\mathbf{X}$, with components $x_{ij} \in \{0,1\}$, $i,j = 1,...,n$, and an n-dimensional vector $\mathbf{y}$, with components $y_j \in \{0,1\}$, $j = 1,...,n$, the following associations are imposed:

$$x_{ij} = \begin{cases} 1, \text{if node } i \text{ is allocated to median } j \\ 0, \text{otherwise} \end{cases}$$

$$y_j = \begin{cases} 1, \text{if node } j \text{ is a median} \\ 0, \text{otherwise} \end{cases}$$

The CPMP formulation as an integer-programming problem can, thus, be given as follows:

$$\min_{\mathbf{X},y} f(\mathbf{X}) \equiv \min_{\mathbf{X},y} \sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij} x_{ij} \cdot \tag{1}$$

subject to,

$$\begin{cases} \sum_{i=1}^{n} x_{ij} = 1, j = 1,...,n \\ x_{ij} \le y_j, i, j = 1,2,...,n \\ \sum_{j=1}^{n} y_j = p \\ \sum_{i=1}^{n} x_{ij}.a_i \le c_j, \text{for } j \text{ such that } y_j = 1 \end{cases} \tag{2}$$

where:

n = number of nodes in the graph
a_i = demand of node i
c_j = capacity of median j
d_{ij} = distance between nodes i and j
p = number of medians to be allocated

After all p medians are chosen, the CPMP becomes a Generalized Assignment Problem (GAP); that is, given a set of medians, allocate a set of clients to those medians so as to minimize Eq. (3):

$$\min_{\mathbf{X}} \sum_{i=1}^{n} \sum_{j=1}^{p} d_{ij} x_{ij} \cdot \tag{3}$$

where d_{ij} is the cost of allocating client i to median j, n is the number of nodes, and p is the number of medians, subject to the capacity of the respective medians and client demands:

$$\begin{cases} \sum_{i=1}^{p} x_{ij} = 1, j = 1,..., p, \\ \sum_{i=1}^{n} x_{ij} \cdot a_i \le c_j, j = 1,..., p \\ x_{ij} \in \{0,1\}, i = 1,..,m; j = 1,.., p \end{cases} \tag{4}$$

where a_i is the demand of client i, and c_j is the capacity of median j.

3 Ant System and Max-Min Ant System

The ant system (AS) [3] was the first ant-based algorithm applied to solve combinatorial optimization problems. More than one decade after it was introduced, several different versions, improvements and applications have been presented (c.f. Dorigo and Di Caro [4], Dorigo and Stützle [6], de Castro and Von Zuben [7]). This section briefly reviews the original proposal together with one of its most popular variants, the Max Min Ant System (MMAS).

3.1 Ant System

The basic AS [3] is conceptually simple, as described in Algorithm 1.

```
Function AS()
   While it < max_it do,
         For each ant do,
            build_solution();
            update_pheromone();
         Endfor
   Endwhile
End
```

Algorithm 1: Pseudocode for the basic ant system (AS).

In Algorithm 1, procedure build_solution() builds a solution to a problem based on a pheromone trail and on optional information heuristics. Each ant k traverses one node per iteration step t and, at each edge, the local information about its pheromone level, τ_{ij}, is used by the ant such that it can probabilistically decide the next node to move to, according to the following rule:

$$p_{ij}^k(t) = \begin{cases} \dfrac{[\tau_{ij}(t)]}{\sum_{j \in J^k} [\tau_{ij}(t)]} & \text{if } j \in J^k \\ 0 & \text{otherwise} \end{cases}, \quad (5)$$

where $\tau_{ij}(t)$ is the pheromone level of edge (i,j), and J^k is the list of nodes yet to be visited by ant k.

While traversing an edge (i,j), ant k deposits some pheromone on it – procedure update_pheromone() – and the pheromone level of edge (i,j) is updated according to Eq. (6).

$$\tau_{ij} \leftarrow \rho.\tau_{ij} + \Delta\tau_{ij}, \quad (6)$$

where $\rho \in (0,1]$ is the pheromone decay rate, and $\Delta\tau_{ij}$ is the increment in the pheromone level. In minimization problems, the pheromone increment is given by

$$\Delta\tau_{ij} = \begin{cases} 1/f(S), & \text{if } (i,j) \in S \\ 0, & \text{otherwise} \end{cases}, \quad (7)$$

where S is the solution used to update the trail, and $f(S)$ is a function that reflects the quality of a solution, i.e., the lower the value the better the quality assuming a minimization problem is being solved.

In our proposal, the pheromone is represented as a vector, instead of as a bi-dimensional matrix as in the classical AS. This is because the algorithm to be described here works by assigning pheromone to vertices and not to edges, as will be further discussed in Section 4.

3.2 Max Min Ant System (MMAS)

An important improvement to the Ant System, called Max Min Ant System (MMAS), was introduced in Stützle and Hoos [8]. In this implementation, the pheromone trail is updated only on the global best and/or local best solutions, instead of on solutions created by every ant. This promotes a better exploitation of the search space, as it favors the solutions in the neighborhood of the global and local bests. Another improvement is the inclusion of upper and lower bounds to the pheromone level (τ_{max} and τ_{min}), thus helping to avoid stagnation. Initially all trail is set to the upper bound in order to favor exploration. As defined in Stützle and Hoos [8], and in Stützle and Dorigo [9], the upper bound is usually chosen to be the maximum value the pheromone can reach at the final iterations. Following from Eq. (6), the maximum value at a given iteration t is:

$$\tau_i(t) = \sum_{j=1}^{t} \rho^{t-j} . \frac{1}{F_{opt}} + \rho^t.\tau_0 . \quad (8)$$

where F_{opt} is the optimal solution, ρ is the pheromone decay rate, and τ_0 is the initial pheromone value. As $\rho < 1$, when t tends to infinity, the pheromone value is limited to

$$\tau_{max} = \frac{1}{1-\rho} \cdot \frac{1}{F_{opt}} . \quad (9)$$

The problem with Eq. (9) is that the optimal solution F_{opt} is usually unknown. To circumvent this difficulty, F_{best}; that is, the fitness of the best solution found so far, is used in place of F_{opt}, as an approximation.

The lower bound is calculated so as to give a τ_{max}/τ_{min} ratio equal to $2n$ (twice the set of candidates to medians), so it is set to $\tau_{min} = \tau_{max}/2n$. On the one hand, this ratio must not be too high, because the probability of selecting a path with low pheromone level would become too small. On the other hand, if the ratio is too low, the probability of selecting a path with high pheromone level would be very close to the probability of selecting a path with low pheromone level.

4 MMAS Applied to the Capacitated p-Medians Problem

This section describes the general methodology used to apply the proposed modified Max Min Ant System to the CPMP [5]. In particular, it is described how the solutions are built by the algorithm, the use of a density-based information heuristics, a local search procedure, a new updating rule for the ant system, and a stagnation control mechanism.

4.1 Building Solutions

The construction of each solution to the CPMP using MMAS is made as follows. Using the probabilistic equation, Eq. (5), each ant sequentially chooses a set of p nodes to become medians among the n candidate nodes. Note that the pheromone level in our proposal is attributed to nodes and not edges. After the definition of the p nodes that will play the role of medians, each one of the remaining $n - p$ nodes has to be allocated to precisely one median, giving rise to the Generalized Assignment Problem (GAP) described in Section 2.

The resulting GAP will not be solved using ant system. Instead, a constructive heuristic to allocate clients to medians will be adopted. The method used here was proposed by Osman and Christofides [1] and works as summarized in Algorithm 2.

```
Function [x] = GAP(clients[ ],medians[ ],n,p)
      ordered_clients = sort_clients();
      For i = 1 to n do,
         ordered_medians = sort_medians(ordered_clients[i]);
         For j = 1 to p do,
            If (capacity(ordered_medians[j]) –
               demand(ordered_clients[i])) >= 0,
                  x[ordered_clients[i]][ordered_medians[j]]=1;
            Endif
         Endfor
      Endfor
 End
```

Algorithm 2: Constructive heuristics to allocate clients to medians.

Function sort_clients() generates a list with all the n clients in increasing order of distance to their corresponding nearest median. Then, the algorithm loops sequentially through this list calling sort_medians() to each client, which generates a list with all the p medians in increasing order of distance to the current client. Given the ordered list of medians, the current client will be allocated to the first available median, i.e., the one for which the difference between median capacity and client demand is greater or equal to zero. The result is the matrix x_{ij} described in Section 2.

After these steps the solution is evaluated by means of Eq. (1). Then, one iteration of local search is performed as described in Section 4.3. This procedure is then repeated for each ant. For stagnation control, if the algorithm does not improve the solution for 30% (defined empirically) of the number of total iterations, all pheromone trails are restarted.

4.2 Information Heuristic (η)

In order to improve the solutions found by the constructive phase of the ant algorithm, an information heuristic which contains the quality of choosing each node as part of the solution is used. For this heuristic, some greedy information concerning the problem is often adopted, such as the distance among the current node to the others in traveling salesperson problems as shown in Dorigo [3], Dorigo and Di Caro [4], and Dorigo and Stützle [6]. Thus, Eq. (5) becomes Eq. (10):

$$p_i^k(t) = \begin{cases} \dfrac{[\tau_i(t)]^{\alpha}.[\eta_i]^{\beta}}{\sum_{l\in J^k}[\tau_l(t)]^{\alpha}.[\eta_l]^{\beta}} & \text{if } i \in J^k \\ 0 & \text{otherwise} \end{cases}, \qquad (10)$$

The parameters α and β are user-defined and control the relative weight of trail intensity $\tau_i(t)$ and information heuristic η_i.

In the case of CPMP, the information heuristic proposed here is a density model for this problem based on Ahmadi and Osman [10]. The idea is to calculate an optimistic density of a cluster if a given node was to be chosen as the median. The computation follows Algorithm 3.

```
Function [η] = density()
  For i = 1 to n do,
    ordered_nodes = sort_nodes(i);
    [all_nodes, sum_distance] = allocate(i,ordered_nodes);
    η_i = all_nodes / sum_distance;
  Endfor
End
```

Algorithm 3: Calculating the density of a cluster.

Function sort_nodes() sorts all nodes based on their distance to node i; and function allocate() assigns each node in ordered_nodes to i, until its capacity is reached, returning two outputs: (i) all_nodes: the number of allocated nodes; and (ii) sum_distance: the summation of the distance between each allocated node and node i. Although this is a reasonable measure of the potential of a node as a candidate to become a median, it does not always imply the most appropriate scenario. Given that information heuristic provides just an approximated indication of the best candidates to become medians, parameters α and β are set so as to emphasize pheromone instead of the heuristic information, as will be observed in the experiments described in Section 5. This is opposed to the approach usually adopted for tackling the TSP problem, in which the best results found are given more importance.

4.3 A Local Search Procedure for the CPMP

The local search heuristic is a first improvement approach for the MMAS algorithm. Basically it consists of changing a client into a median and this median into a client seeking an improvement of the objective function.

To define the search neighborhood, an approach based on the uncapacitated p-medians problem proposed in Resend and Werneck [11], and in Teitz and Bart [12] was adopted. This approach consists of an optimistic function to calculate a *profit*, P, obtained by changing a client into a median and determining which median to remove in this case. Initially, two vectors d_1 and d_2, containing the first and second closest median to each client, are calculated. Then, the profit associated with each possible interchange between a client and a median obeys the following equation:

$$P(f_i, f_r) = \sum_{u:d_1(u)\neq f_r} \max\big(0,[d_1(u) - d(u, f_i)]\big) - \sum_{u:d_1(u)= f_r} \big(\min(d_2(u), d(u, f_i)) - d_1(u)\big), \qquad (11)$$

where M is the set of medians, $f_i \notin M$ and $f_r \in M$ is applied, f_i is the node chosen to enter the solution as a median, f_r is the node chosen to leave it, $d_1(u)$ is the distance of node u to its nearest median, $d_2(u)$ is the distance of node u to its second nearest median, and $d(x_1,x_2)$ is the Euclidean distance between x_1 and x_2.

Figure 1 provides a general overview of the local search procedure. The larger circles represent candidates to medians, and the smaller ones represent the clients.

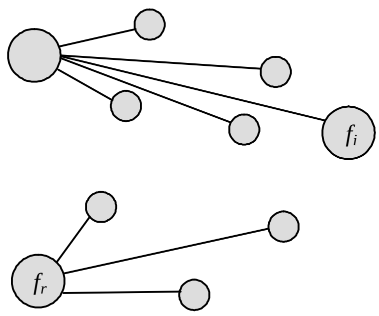

Figure 1: Overview of the local search procedure. Solid lines connect clients to medians, f_i is the candidate to enter the solution as a median, and f_r is the median that will leave the solution.

The first term of the right hand side of Eq. (11), detailed in Figure 2, refers to all clients that do not belong to the median candidate to leave the solution, and takes into account two possibilities: (i) the new median is nearer to the client than its previous nearest median; or (ii) the new median is farther to the client than its previous nearest median. When the former holds, allocating this client to the new median will reduce the total value of the objective function, increasing the *profit* in proportion to $d_1(u) - d(u,f_i)$. Otherwise, no change occurs.

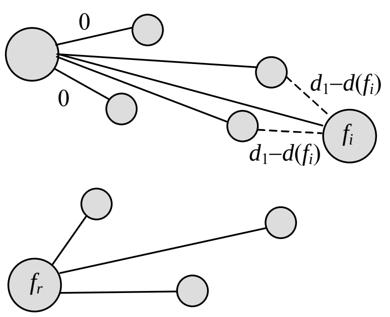

Figure 2: First term of the right hand side of Eq. (11). Clients not allocated to f_r that will profit from the insertion of f_i. Solid lines connect each client to a median, dotted lines represent the new connections, f_i is the candidate to enter the solution as a median, and f_r is the median that will leave the solution.

In the second term of the right hand side of Eq. (11), those clients that will lose their nearest median and will be allocated to a new one are taken into account (Figure 3). There are also two possibilities in this case: (i) the nearest median becomes the new median; or (ii) the client is allocated to its second nearest median. In the first case, the difference on the profit will be $d(u,f_i) - d_1(u)$, and can make the total distance larger or smaller. In the second case, the profit function will have a decrease proportional to $d_2(u) - d_1(u)$.

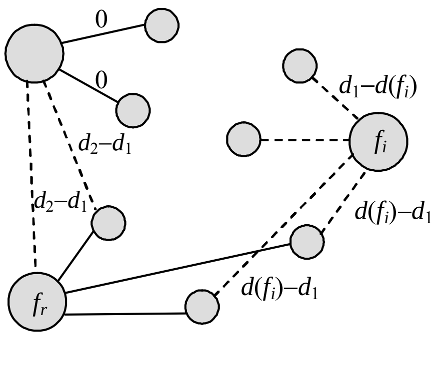

Figure 3: Second term of the right hand side of Eq. (11). Clients allocated to f_r that will profit or not from the exit of f_r. Solid lines connect each client to a median, dotted lines represent the new connections, f_i is the candidate to enter the solution as a median, and f_r is the median that will leave the solution.

After this procedure has finished, two similar First Improvement Local Search procedures are performed, but regarding the clients instead of medians. The first one (Figure 4(a)) consists of interchanging two clients of different medians whenever it is profitable, and after that, recalculating the medians taking the best point inside each cluster. The second type (Figure 4(b)) is the same as the previous one, but two clients of one median are interchanged with one client from another.

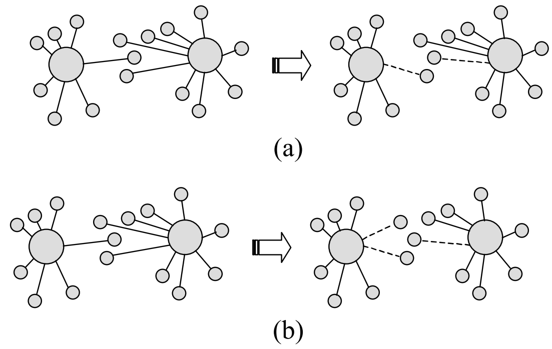

Figure 4: First Improvement Local Search procedures. (a) One movement of 1-interchange GAP local search. One client of the first cluster interchanges with a client from the other one (dotted lines). (b) One movement of 2-interchange GAP local search. One client of the first cluster interchanges with two clients from the other one (dotted lines).

A number greater than 2 for the λ-interchange algorithm is computationally expensive and will hardly represent significant benefits to the final results, because 1- and 2-interchange, when tried several times, can eventually perform the job of a λ-interchange for higher values of λ.

4.4 A New Updating Rule for AS

A well-known problem with the AS is that of scaling the objective function to update the pheromone trail. If not appropriately done, the performance of the algorithm tends to be unsatisfactory for large instances of the problem. To propose a suitable updating rule, a framework for the AS that can also be applied to its variants like MMAS was introduced in Blum *et al.* [13] and Blum and Dorigo [14]. The main idea is to normalize the bounds of the pheromone trails in the range [0,1], and consequently normalize the quality function of a solution, $f(\cdot)$. The updating rule thus becomes:

$$\tau_i = \tau_i + \rho(\Delta\tau_i - \tau_i). \qquad \textbf{(12)}$$

with

$$\Delta\tau_i = \begin{cases} \dfrac{1/f(S_{best})}{\sum_j 1/f(Sj)}, & \text{if } i \in S_{best} \\ 0, & \text{otherwise} \end{cases}. \qquad \textbf{(13)}$$

for each node *i*, where S_{best} is the best solution found and S_j is the solution found by ant *j*. In this case, τ_i, $\forall i$, is initially set to 0.5 in order to equal the chances in both directions.

4.5 Adaptation to the CPMP

To apply the MMAS to the capacitated *p*-medians problem using the new updating rule, some modifications had to be introduced to take full advantage of all the problem information available. First, $\tau_{\min}$ and $\tau_{\max}$ were set to 0.001 and 0.999, respectively, and the pheromone trail was initialized to 0.5.

It can be noticed from Eq. (12) that in order to have a positive increase in the pheromone level it is necessary that $\tau_t < \Delta\tau_t$, and, as the solutions obtained per iteration have a value near the best so far, Eq. (13) will hardly produce a value greater than 0.5, so there is a bound lower than $\tau_{\max}$ imposed by $\Delta\tau_t$.

For this reason it is proposed a new updating rule, presented in Eq. (14), where a simpler calculation taking into account just the two solutions used to update the pheromone is made, giving a better quality function.

$$\Delta\tau_i = \begin{cases} 1 - \dfrac{l_{best} - g_{best}}{l_{worst} - l_{best}} & i \in \{g_{best}, l_{best}\} \\ 0 & \text{otherwise} \end{cases} . \quad \textbf{(14)}$$

where g_{best}, l_{best}, and l_{worst} are the global best, local best and local worst solutions, respectively. In Eq. (14), whenever the local best is better than the global one, the pheromone is updated proportionally to the difference between them. Otherwise, the complementary value is taken; thus, the closer the local best from the global best, the closer $\Delta\tau_t$ becomes to one. Note that the global best information is updated after this process, so Eq. (14) can result in a range [0,2], meaning that whenever a new best solution is found, the vertices with low pheromone values are set to a value near or equal to $\tau_{\max}$ and those which are already with a high value will be equal to $\tau_{\max}$.

4.6 Pheromone Stagnation Control

A stagnation control mechanism for the algorithm is proposed so that it is restarted every time it stagnates. For this problem, it is intuitive that when the pheromone trail converges, *p* points (number of medians) will be at the upper bound, $\tau_{\max}$, and the remaining will be at the lower bound, $\tau_{\min}$. Thus, every time the sum of all pheromone follows Eq. (15), the algorithm is said to have stagnated and is thus restarted:

$$\sum_i \tau_i = p \cdot \tau_{\max} + (n - p) \cdot \tau_{\min} . \quad \textbf{(15)}$$

5 Performance Evaluation

To evaluate the performance of the modified algorithm, several CPMP instances from the literature were tested. For each instance of a given set, it was made the calculation of the *relative percentage deviation* from the best known solution: $RPD = 100 \times (S_{\text{MMAS}} - S_{best})/S_{best}$ and then the average was taken, where S_{MMAS} is the best solution found by MMAS and S_{best} the best solution known for each instance.

The first experiment was performed to assess the influence of the information heuristics on the MMAS. Simple experiments were run on the classic instances of Osman [1] and Lorena [15]. Table 1 presents some results found with two sets of parameters α and β to illustrate the importance of the heuristic information. 500 iterations of the improved MMAS (IMMAS) were run with the following parameters: $\alpha = 1$, $\beta = 0$ (only pheromone and no heuristic information), and $\alpha = 3$, $\beta = 1$ (with heuristics but privileging pheromone). As can be seen from Table 1, the heuristic information successfully improves the results found by the Ant System (i.e. only using pheromone information).

Table 1: Average relative percentage deviation from the best known solution to the Osman and Lorena sets. Influence of the information heuristics. Negative results mean that a solution better than the best solution found so far was found

		$\alpha = 1, \beta = 0$	$\alpha = 3, \beta = 1$
Osman	Average (%)	0.081203	**0.064181**
Lorena	Average (%)	0.090277	**–0.11838**

To illustrate the performance of the two different pheromone updating rules studied, some experiments were performed with the IMMAS algorithm applied to the same instance sets as above for 500 iterations. As can be seen from Tables 2 and 3, on harder instances (i.e. larger number of clients and medians to search and harder GAP instances generated from each *p*-median solution) Eq. (14) gives better results than Eq. (13). On easier ones, they both give the same results.

Table 2: Comparison between the two pheromone updating rules in 500 iterations on the first instance set. The first group of columns corresponds to the first 10 instances of Osman [1] and the second group represents the last 10 instances. The best results are presented in bold. "Δτ" represents the solution obtained using Eq. (13) while "new Δτ" represents the solution obtained using Eq. (14). "*n*" represents the number of clients and "*p*" represents the number of medians for each instance.

	Δτ	new Δτ		Δτ	new Δτ
Osman	**Sol.**	**Sol.**		**Sol.**	**Sol.**
***n*=50 *p*=5**	**713**	**713**	***n*=100 *p*=10**	1008	**1007**
	740	**740**		**966**	**966**
	751	**751**		**1026**	**1026**
	651	**651**		**983**	**983**
	664	**664**		**1091**	**1091**
	778	**778**		**955**	**955**
	787	**787**		**1034**	**1034**
	820	**820**		**1043**	**1043**
	715	**715**		**1032**	**1032**
	831	**831**		1007	**1005**

Table 3: Comparison between the two pheromone updating rules in 500 iterations on the second instance set. The best results are presented in bold.

	Δτ	new Δτ		Δτ	new Δτ
Lorena	**Sol.**	**Sol.**		**Sol.**	**Sol.**
***n*=100 *p*=10**	17377	**17352**	***n*=300 *p*=30**	41228	**40790**
***n*=200 *p*=15**	**33254**	**33254**	***n*=402 *p*=30**	63966	**62400**
***n*=300 *p*=25**	**45279**	**45279**	***n*=402 *p*=40**	53909	**52857**

In the next experiments, comparisons were performed between MMAS and IMMAS. For each algorithm, 10 trials of 2,000 iterations were run on an Athlon XP + 2000, 1.67GHz, 512 MB RAM running Slackware 9.1, compiled with gcc 3.2, not optimized at compilation.

Error! Reference source not found. presents the set of problems first introduced in Osman and Christofides [1] and broadly studied in the CPMP literature. The data sets are available at the OR-Library (http://www.brunel.ac.uk/depts/ma/research/jeb/info.html), a repository of test data sets for a variety of Operations Research (OR) problems.

The results were compared with those presented in Osman and Christofides [1], referred to as HSS.OC, which is an implementation of a hybrid involving Simulated Annealing and Tabu Search. As can be observed from this table, the IMMAS algorithm performed better than the MMAS alone and is competitive when compared to the HSS.OC algorithm. It must also be noticed that the variance of the solutions found was 0%, meaning that the same results were found for all 10 trials, indicating the robustness of the algorithm.

Error! Reference source not found. shows the set of problems created by Lorena in [15], where geographical information about a large city in Brazil was obtained, thus creating a more realistic and complex scenario. In this particular problem, the IMMAS presents a superior performance when compared with the simple MMAS algorithm. Furthermore, IMMAS was capable of finding better solutions than the best solutions known to date. It is also important to observe that the most noticeable differences in performance are on the larger instances, thus suggesting that the proposed modifications help to overcome one important difficulty of Ant Systems, associated with problems containing a large dataset.

6 Discussion and Future Trends

This paper presented the application and further improvements of an ant-colony optimization algorithm to the capacitated *p*-medians problems (CPMP). In particular, it described one form of applying the Max Min Ant

Table 4: MMAS, IMMAS and HSS.OC results for Osman's set of instances, the "Best" column is the best known solution found so far, "Sol." is the solution obtained by each algorithm, "%" is the average relative percentage deviation from best and "Time" is the execution time in seconds, results in boldface are the best found by comparing the algorithms.

		MMAS			IMMAS			HSS.OC	
Osman	**Best**	**Sol.**	**%**	**Time(s)**	**Sol.**	**%**	**Time(s)**	**Sol.**	**%**
***n*=50 *p*=5**	713	**713**	0.00	31.93	**713**	0.00	28.22	**713**	0.00
	740	**740**	0.00	33.52	**740**	0.00	30.11	**740**	0.00
	751	**751**	0.00	45.59	**751**	0.00	37.83	**751**	0.00
	651	**651**	0.00	47.51	**651**	0.00	32.73	**651**	0.00
	664	**664**	0.00	40.24	**664**	0.00	31.98	**664**	0.00
	778	**778**	0.00	39.88	**778**	0.00	33.37	**778**	0.00
	787	**787**	0.00	44.00	**787**	0.00	34.19	**787**	0.00
	820	822	0.24	56.31	**820**	0.00	36.95	**820**	0.00
	715	**715**	0.00	44.05	**715**	0.00	33.64	**715**	0.00
	829	831	0.24	49.12	**829**	0.00	40.12	**829**	0.00
***n*=100 *p*=10**	1006	1008	0.20	316.03	1007	0.09	158.34	**1006**	0.00
	966	**966**	0.00	180.66	**966**	0.00	156.33	**966**	0.00
	1026	**1026**	0.00	180.56	**1026**	0.00	168.29	**1026**	0.00
	982	985	0.30	152.64	**982**	0.00	194.13	985	0.31
	1091	1092	0.09	118.62	**1091**	0.00	154.32	**1091**	0.00
	954	955	0.10	120.84	955	0.10	186.21	**954**	0.00
	1034	**1034**	0.00	150.60	**1034**	0.00	162.23	1039	0.48
	1043	**1043**	0.00	142.65	**1043**	0.00	167.21	1045	0.19
	1031	1033	0.19	118.23	1032	0.09	164.43	**1031**	0.00
	1005	1009	0.40	178.15	**1005**	0.00	214.39	**1005**	0.00
Avg.			0.088			**0.014**			0.049

Table 5: MMAS and IMMAS for Lorena set of instances, the "Best" column is the best known solution found so far, results in bold are the best found among the algorithms, and "Time" is the execution time in seconds, for IMMAS the column "Sol." holds the average result in 10 trials and standard deviation in parentheses, and the "Best" column represents the best solution found along these trials.

		MMAS			IMMAS			
Lorena	**Best**	**Sol.**	**%**	**Time**	**Sol.**	**Best**	**%**	**Time**
***n*=100** / ***p*=10**	17288	17288	0.00	295.95	17264.6 (17.08)	17252	**–0.21**	253.77
***n*=200** / ***p*=15**	33395	33254	–0.42	540.96	33203.7 (12.98)	33187	**–0.62**	1428.10
***n*=300** / ***p*=25**	45364	45251	–0.25	8109.64	45279.1 (32.15)	45245	**–0.26**	1742.26
***n*=300** / ***p*=30**	40635	40638	0.01	7818.13	40662.5 (116.54)	40521	**–0.28**	2127.22
***n*=402** / ***p*=30**	62000	62423	0.68	12701.24	62280,75 (187,24)	62020	**0.03**	1407.44
***n*=402** / ***p*=40**	52641	52649	0.02	10500.15	52627.5 (108.82)	52492	**–0.28**	1484.44
Avg.			0.006				**–0.271**	

System (MMAS) to the CPMP problem that includes a local search heuristics, and combines the MMAS with a new updating rule and a recent framework from the literature in order to improve the performance of the algorithm, mainly when large instances are considered. It is an extension of a previous work by the authors [5].

With the extensions proposed here, based on the framework presented in Blum *et al.* [13] and Blum and Dorigo [14], the results obtained showed that the modified algorithm is competitive and sometimes better than other heuristics found in the literature when applied to the same problem instances. It could also be noted that, over ten trials, the variance in the behavior of the modified algorithm was very small, sometimes zero, for the smaller instances.

Despite the quality of the results already achieved, there are some important aspects that deserve further investigation. For instance, even though the density function gives information about promising medians, it is only accurate on the choice of the first points, because, as a point is chosen, the density surface of the search space changes accordingly. So, recalculating this value for all the candidates not yet chosen could improve the quality of the results obtained, but with the drawback of a high computational cost. Furthermore, it must also be investigated an adaptive distribution of the importance factors given for the pheromone and information heuristic (α and β) so as to try to improve the performance of the algorithm. This happens because, initially, the algorithm has no information about the pheromone trail. Thus, a higher importance should be given to η during the first iterations, and after a number of iteration steps the pheromone trail can give better information than η, so it must have a higher importance as well. Finally, a better GAP local search, or even a constructive heuristics, can be implemented to further improve the assignment of clients to medians.

Acknowledgements

The authors thank CAPES, FAPESP and CNPq for the financial support, and the reviewers for their relevant comments and suggestions.

References

[1] OSMAN, I. H.; CHRISTOFIDES, N., Capacitated Clustering Problems by Hybrid Simulated Annealing and Tabu Search. Pergamon Press, England, 1994, Int. Trans. Opl. Res. v. 1, n. 3, pp. 317-336, 1994.

[2] GAREY, M.R. & JOHNSON, D. S., Computers and Intractability: A Guide to the Theory of NP-Completeness. San Francisco: Freeman. 1979.

[3] DORIGO M. Optimization, Learning and Natural Algorithms. Ph.D.Thesis, Politecnico di Milano, Italy, in Italian, 1992.

[4] DORIGO, M. & DI CARO, G. The Ant Colony Optimization Meta-Heuristic. In D. Corne, M. Dorigo and F. Glover (Eds.), *New Ideas in Optimization*. McGraw Hill, London, UK, Chapter 2, pp. 11-32. 1999.

[5] DE FRANÇA, F. O., VON ZUBEN, F. J., DE CASTRO, L. N. Definition of Capacited p-Medians by a Modified Max Min Ant System with Local Search In: ICONIP - 2004 11th International Conference on Neural Information Processing - SPECIAL SESSION ON ANT COLONY AND MULTI-AGENT SYSTEMS, 2004, Calcutta. Lecture Notes in Computer Science. v. 3316. pp. 1094 – 1100, 2004.

[6] DORIGO, M. & STÜTZLE, T. The Ant Colony Optimization Metaheuristic: Algorithms, Applications, and Advances. In F. Glover and G. A. Ko-

chenberger, *Handbook of Metaheuristics*, Kluwer Academic Press, Chapter 9, pp. 251-286. 2003.

[7] DE CASTRO, L. N. & VON ZUBEN, F. J. (eds.) *Recent Developments in Biologically Inspired Computing*, Idea Group Inc. 2004.

[8] STÜTZLE, T. & HOOS, H.H. The MAX-MIN ant system and local search for the traveling salesman problem. In T. Bäck, Z. Michalewicz, and X. Yao, editors, Proceedings of the IEEE International Conference on Evolutionary Computation (ICEC'97), IEEE Press, Piscataway, NJ, USA, pp. 309-314, 1997.

[9] STÜTZLE T. & DORIGO M. ACO algorithms for the Quadratic Assignment Problem. In D. Corne, M. Dorigo, and F. Glover, editors, New Ideas in Optimization, pp. 33-50. McGraw-Hill, 1999.

[10] AHMADI, S. & OSMAN, I.H., Density based problem space search for the capacitated clustering problem. Annals for Operational Research, 2004 (in press).

[11] RESENDE, G.C.M. & WERNECK, F. R. On the implementation of a swap-based local search procedure for the p-median problem. Proceedings of the Fifth Workshop on Algorithm Engineering and Experiments (ALENEX'03), Richard E. Ladner (Ed.), SIAM, Philadelphia, pp. 119-127, 2003.

[12] TEITZ, M. B.; BART, P., Heuristic Methods for Estimating the Generalized Vertex Median of a Weighted Graph. Operation Research, 16(5):955-961, 1968.

[13] BLUM, C., ROLI, A. & DORIGO, M. HC-ACO: The hyper-cube framework for Ant Colony Optimization. In Proceedings of MIC'2001 - Metaheuristics International Conference, v. 2, Porto, Portugal, pp. 399-403, 2001.

[14] BLUM, C. & DORIGO, M., The Hyper-Cube Framework for Ant Colony Optimization. IEEE Transactions on Systems, Man and Cybernetics, Part B, 34(2): 1161-1172, 2004.

[15] LORENA, L.A.N. & SENNE, E.L.F. Local Search Heuristics for Capacitated P-Median Problems, Networks and Spatial Economics 3, pp. 407-419, 2003.