Assigning cells to switches in mobile networks using an ant colony optimization heuristic

Joseph R.L. Fournier, Samuel Pierre*

Abstract

The problem of assigning cells to switches in a mobile network is an NP-Hard problem. It is therefore necessary to use a heuristic method to solve it in a reasonable amount of time. This paper applies the relatively new ant colony optimization meta-heuristic to this problem. It has also hybridized the local optimization $k$-opt technique with the heuristic. The developed algorithms have been applied to both problems (single and dual homing) described by the first studies regarding the assignment problem. In comparison to previous research, our methods are relatively efficient, regarding both quality of solutions provided and execution time of algorithms. It would thus be possible to increase the size of problem instances without augmenting the algorithm’s execution time by a factor larger than $n^2$ and obtain solutions with an equivalent quality. The algorithm implemented for the double homing problem could be used to determine the best network configuration according to the number of known or anticipated calls for each cell, for each considered period of the day. It would afterwards be possible to design a dynamic algorithm that would take into account the existing physical links, which would determine in real time the appropriate moment for changing assignment (in respect of their management) of cells having two links, thus reducing the total number of handoffs occurring in the network.

Keywords: Optimization; Heuristic; Assignment of cells to switches; Mobile network

1. Introduction

The number of users of mobile telecommunication services has not ceased to increase in the last decades. Meanwhile, the integration of new services to mobile networks intensifies the strain exerted on the resources of these networks. Whilst having obvious economic, intellectual and social advantages, this diversification of services brings numerous technological and design challenges for developers. In addition, because of budgetary restrictions imposed upon companies responsible for the implementation and administration of existing and future mobile networks, these providers need to diminish as much as possible their infrastructure and operational costs in order to remain competitive. The present paper addresses one of the means of reducing these costs: finding an efficient way to assign cells to the switches in the planning phase of mobile networks.

Many components enable mobile networks to provide services similar to ones offered by traditional networks. The mobile telephone switching office (MTSO), which comprises the mobile switching center (MSC) or switch, is instrumental in relaying the calls from its assigned cells to the conventional network’s (PSTN—public switched telephone network) interface. The geographical zone covered by a mobile network is divided into cells, each of which has a base transceiver station (BTS) in charge of managing communications occurring on its ‘territory’. The BTS contains an antenna enabling it to transfer (with
the help of a ‘transmitter’) and to receive (with the help of a ‘transceiver’) radio frequencies exchanged between itself and the mobile subscriber units (MSU) present in its cell. This cell configuration has been devised in order to be able to reuse frequencies in several non-adjacent cells. Each cell is meanwhile assigned to a switch. A problem occurs when a user moves from one cell to a neighbouring cell. Since two adjacent cells cannot use the same frequency, any occurring call must be transferred from one frequency (used in the cell from which the user originates) to another one (used in the cell in which the user is moving). This frequency change is called handoff.

We distinguish two types of handoff: simple handoff in which only one switch is involved, and complex handoff in which at least two switches are involved. A complex handoff consumes more network resources than a simple handoff since the user-location database must be updated after the user moves. Furthermore, since the information needed to invoice the user will most likely be compiled via the first switch, it will not be possible to disconnect him from this switch and therefore, the call will have to transit through both switches involved.

The assignment of cells to switches is an NP-Hard problem, having an exponential complexity (i.e. \( n \) cells need to be assigned to \( m \) switches). One must therefore use a heuristic approach to solve the problem in a reasonable time. We have applied the relatively new meta-heuristic of ant colony optimization (ACO) to this problem in order to analyze its effectiveness compared to other methods. The heuristic has also been combined with the local optimization \( k \)-opt technique so as to determine if this combination would improve results. Several variants of the methods have been created and these algorithms have been applied to the two problems (single and double homing), which were initially described by Merchant and Sengupta [14,15].

The rest of the paper is organized as follows. Section 2 states the assignment problem and summarizes the different heuristics that have been used to solve it. Section 3 describes the methods used and provides details regarding their implementation. The results of our algorithms are analyzed in Section 4. Finally, Section 5 provides a summary of our work, outlines its limitations and indicates possible future improvements.

2. Background and related work

Given a number of cells and switches, whose respective positions are known, the assignment problem consists of assigning each cell to a switch of the network so as to approach the optimum as much as possible. The objective is to minimize implementation and operational costs. Our model considers two types of cost: the link cost regarding cells to switches and the complex handoff cost between cells. The simple handoff cost is not directly taken into account since it is usually negligible in comparison to the complex handoff cost, it is unavoidable and can be considered as a constant.

Merchant and Sengupta [14,15] were the first authors to clearly define the cell assignment problem and to develop a heuristic in order to solve it. They devised an algebraic formulation of the problem as a linear integer program. Suppose the network has \( n \) cells which need to be assigned to \( m \) switches. Let us define variable \( h_{ij} \) as the cost, per time unit, of a complex handoff between cells \( i \) and \( j \), each cell being assigned to a different switch. The value of the variable \( h_{ij} \) varies according to the number of handoffs occurring, per time unit, between the cells in question. Variable \( c_{ik} \) is defined as the amortization cost, per time unit, of the link between cell \( i \) and switch \( k \) \((i=1,\ldots,n; k=1,\ldots,m)\). Let us also define the binary variables \( x_{ik} \) as being equal to 1 if cell \( i \) is assigned to switch \( k \) and equal to 0 if not. Meanwhile, the binary variables \( y_{ij} \) are for their part equal to 1 if cells \( i \) and \( j \) are both assigned to the same switch and equal to 0 if assigned to different switches. Finally, the number of calls, per time unit, intended for cell \( i \) is represented by variable \( \lambda_i \), while a switch’s capacity is symbolized by \( M_k \). Each cell must be assigned to only one switch and a switch’s capacity cannot be violated, as indicated by the following formula:

\[
\sum_{i=1}^{n} \lambda_i x_{ik} \leq M_k \quad \text{for } k = 1, \ldots, m
\]

(1)

The objective-function is the following:

\[
\min f = \sum_{i=1}^{n} \sum_{k=1}^{m} c_{ik} x_{ik} + \sum_{i=1}^{n} \sum_{j=1, i \neq j}^{n} h_{ij}(1 - y_{ij})
\]

(2)

The first term of the function adds the link costs and the second the complex handoff costs. (Refer to Refs. [14,15] for a more detailed formulation of the problem.) Merchant and Sengupta [14,15] have also provided a formulation of the double homing variant of the problem in which there are two periods per day, during which the volume of calls per cell and the handoff patterns are different. The authors have applied their algorithms to the two variants of the problem (single and double homing). To solve the double homing problem, they apply the same algorithms used to solve the single homing variant to the two periods of the problem. The instances of the two periods are sequentially solved, taking care not to count link costs twice if a cell is assigned to the same switch for the two periods.

Hedible et al. [10] proposed a genetic algorithm (GA) to solve the problem. Their implementation gave better results than those generated by Merchant and Sengupta’s heuristics. Houéto and co-workers [11,12,16] applied the tabu search (TS) technique to the problem. The results obtained by this heuristic proved to be better than those of the GA mentioned above. The constraints programming (CP) method has been used by Amoussou et al. [1,3] and André et al. [2]. Amoussou et al. [1,3] developed an exact algorithm that can be used to solve small-scale problems
(e.g. 50 cells and 4 switches). André et al. [2] improved this exact method by reducing its execution time and have also conceived another algorithm (for instances of all sizes), which hybridizes their own TS implementation with CP. This algorithm has produced results that are close to optimum. Their different implementations aim at reducing the objective-function’s global cost by means of assignment modifications (redistributions, double reassignments and ejection-chain type reassignments). This heuristic has proved to be the best of all techniques used to solve this problem. André et al. [2] have also adapted their algorithm to the double homing problem. The results generated by this implementation are less economical than those found with the single homing variant, the authors having made a somewhat simplistic adaptation of their algorithm.

3. Proposed method

3.1. ACO meta-heuristic

The ACO method constitutes, like the TS technique, a meta-heuristic generally used to solve combinatorial optimization problems considered to be NP-Hard. It was proposed by Dorigo and Di Caro [8] (see also Dorigo et al. in Ref. [7]) and was based on a simpler method called ant systems (AS) created by Dorigo [4–6].

The technique was inspired from the research of Goss et al. [9] in which real ants from Argentina were used. The experiment, performed in a laboratory, demonstrated that after a transitory phase, all ants of a colony systematically took the shorter path amongst two possible alternatives to travel between their nest and a source of food. The selection of the shortest branch can be explained by an autocatalysis phenomenon, which takes the form of an indirect communication called stigmergy that influences the local environment. The ants deposit on the ground a chemical substance named pheromones while traveling. Upon arrival at an intersection, ants make their choice of the path to follow according to a probability that is biased by the quantity of pheromones on each trail. Initially, those ants that happen to have followed the shortest path will arrive first at the food source. During their travel back to the nest, they will sense the deposited pheromones on the shortest path (which were deposited by themselves) and therefore will choose it with a higher probability than the other longer one, which does not yet contain any pheromone at the intersection (the ants which will have followed the longer path will not have reached the intersection at the time the shortest path ants return to the nest). More pheromones will then be deposited on the shortest track, inciting other ants to select it instead of the longer route. Thus, the more pheromones on the shortest path, the more ants will be encouraged to select it, thus increasing the quantity of pheromones added to it. After a while, the whole colony will solely follow the shortest path between the nest and the food.

During execution of the ACO algorithm as applied to the cell assignment problem, each ‘ant’ constructs its solution in a probabilistic way by iteratively adding components to its partial solution until a final solution is completed. While building their solution, ants take into account the information related to the instance being solved and the experience acquired collectively via the pheromones. When an ant finds itself with a partial solution containing a series of assigned cells, it chooses to integrate the next component (selection of cell not yet assigned) to its partial solution by taking into account data regarding the problem and the pheromone trails concerning the desirability to proceed to particular assignments, for each cell not yet assigned, based on the present state of the partial solution. After having selected the next cell to assign, the ant must choose a switch to assign it to, here again by taking into account data of the problem (link and handoff costs) and pheromone trails, but regarding in this case the desirability of assigning the cell in question relative to each available switch.

An ant selects the next cell to be assigned by using the following pseudo-random proportional rule:

\[
\text{value} = \arg \max_{i \in \mathbb{N}} k(\tau_{ij}(t))^\alpha (\eta_{ij})^\beta
\]

and with probability \(1 - q_0\) randomly select a cell from all cells not yet assigned, where \(\eta_{ij}\) is a selection criterion (hereafter explained), \(\tau_{ij}(t)\) corresponds to the quantity of accumulated pheromones related to an assignment at time \(t\), \(\alpha\) and \(\beta\) are parameters which determine the relative influence of pheromones and selection criterion, and \(N^k\) constitutes the permissible neighbourhood of ant \(k\), composed of all cells not yet affected.

When the fractional value of parameter \(q_0\) approaches 1, the algorithm favors the deterministic use of problem data and pheromone trails while the heuristic implements a random methodology when the value nears 0. Meanwhile, variables \(\alpha\) and \(\beta\) are parameters used to modify the relative influence of information known at the start of algorithm (\(\eta_{ij}\)) and of knowledge acquired (\(\tau_{ij}\) corresponding to the quantity of pheromones at time \(t\)) during execution. The value attributed to \(\eta_{ij}\) regarding problem data is determined by applying the least-regret rule used by Amoussou et al. [1], and André et al. [2] in their CP implementations, regarding order of selection of variables (both link and
handoff costs are taken into account for calculations). The least-regret rule consists of selecting in priority, cells for which the difference between the costs of their two least-costly switches is the highest, reducing thus the risk of having to perform expensive assignments subsequently. It is to note that this rule is weighted by quantity of pheromones deposited by previous ants and by the value of variable $\beta$. Finally, the second rule implemented by André et al. [2], consisting of sorting cells to be initialized in ascending order of their call volume, has been used together with the first one.

The pseudo-random proportional rule defined above is also used to determine the switch to which the selected cell is assigned. The value of formula value $= \arg \max_{\text{switches}} \{(\tau_{ij})^\alpha * (\eta_{ij})^\beta\}$ will in this case determine the desirability of assignment of the cell for each switch $j$. The value of $\eta_{ij}$ is here defined as the assignment cost of cell $i$ to switch $j$. It is calculated by taking into account the link and handoff costs. The assignment having the least cost among possible ones will have the highest $\eta_{ij}$ value (the decreasing $\eta_{ij}$ value of other assignments will be in inverse proportion to cost increase). $\tau_{ij}$ corresponds here again to the quantity of accumulated pheromones related to an assignment, $\alpha$ and $\beta$ to parameters which determine the relative influence of pheromones and of assignment criterion and $k$ to the permissible neighbourhood of ant $k$, composed of all switches having the available capacity.

The algorithm offers the user two alternative routines to calculate assignment costs ($\eta_{ij}$): the least-cost method and the procedure we call the least-increase technique, which determines the value of variable $\eta_{ij}$ in a similar way to that of the first alternative, but applying a weighting factor slightly more important to handoff costs relative to link costs. We have implemented this strategy in order to analyze its effectiveness when applied to data having relatively important handoff costs compared to link costs. This alternative uses all previously defined parameters except for the weighting factors $\alpha$ and $\beta$ regarding assignment. These coefficients can nevertheless be used for the selection of cells to be assigned. The goal of this simplified implementation was solely to evaluate the results of a different approach. It must be noted that this least-increase technique must be used in conjunction with our local search $k$-opt technique that does not authorize a temporary capacity violation. The least-cost method can, for its part, be used with all three implemented variants of our local search heuristic.

After an ant has finished assigning all cells of the network, it deposits pheromone trails on all components (assignments) contained in its solution. There exist several strategies, which can be used regarding updating of pheromones. The egalitarian tactic consists of authorizing all ants to deposit a certain amount of pheromones on their solution components. The right to deposit pheromones can, on the contrary (in an elitist tactic), be granted solely to the ant having obtained the best solution during a cycle. It is also possible to grant a right to deposit pheromones to all ants, but to allow the ant having the best solution to deposit more pheromones than its associates. The flexibility of our implementation enables the use of any of these strategies. In addition, our algorithm gives the possibility of applying a weighting factor to both deposit types (the deposit made by all ants and the deposit made by privileged ants). The value of the coefficients provided by the user determines if the algorithm is executing an egalitarian or an elitist strategy.

There are several other parameters that can be adjusted in order to increase the search efficiency of the algorithm. These include the updating moment of pheromones by ants, the moment during execution of partial or total evaporation of pheromones that have been deposited on components, the number of ants used and the permissibility or prohibition of momentary capacity constraint violations. Our implementation gives the user the possibility to indicate if the ACO heuristic must always comply with the switches’ capacity constraints or if it can temporarily violate them. All final solutions must, however, satisfy capacity constraints. When the user prohibits any violation (even temporarily), he shall either execute the ACO heuristic alone or with the first local search $k$-opt variant implemented, whose objective is to try to improve the satisfactory solution provided by the ACO heuristic. When he authorizes momentary capacity violations, he must either use the second or the third $k$-opt variant, whose purpose is to restore capacity constraints, using an approach similar to the standard $k$-opt method.

The $k$-opt method (where $k \geq 2$) was developed by Lin and Kernighan [13]. It consists of iteratively applying $k$ alterations to a given solution in order to try to improve the objective-function value. This method is widely used and generally produces good results, especially the 2-opt and 3-opt variants, which consists of respectively applying 2 and 3 modifications. The technique is frequently used in conjunction with meta-heuristics as a local search method, enabling one to improve solutions obtained by global heuristics, as in our implementation. Since the main objective of our study is to analyze efficiency of the ACO heuristic applied to our problem, we do not explain the approach used by the three alternatives that we have implemented. It will suffice to note that our implementations apply several types of modifications (regarding assignments) to a given solution.

Finally, our algorithm has recourse to cycles and loops (their quantity being determined by the user) in order to balance search between intensification and diversification. Cycles enable the algorithm to intensify its search in the vicinity of certain regions of search space while loops enable it to diversify its search by eliminating all pheromones (evaporation phenomenon) previously deposited and by directing the search towards new areas at loop start.

Our implementation allows us to adjust the value of all parameters defined above in order to produce the best possible results. We have implemented several variants of our ACO meta-heuristic and of our local $k$-opt search.
The algorithm has been developed in C++ and executed under WIN2000 on a 350 MHz Pentium machine with 96 MB of RAM. The data used for tests were those used by André et al. [2]. Fig. 1 displays the general algorithm.

### 4. Results

#### 4.1. ACO meta-heuristic results

We observe that our implementation of the ACO heuristic ‘suffers’ from the data used for tests of the single homing problem. Recall that the data used for tests were those generated by André et al. [2]. We have used the same data in order to be able to compare our results with those of these authors. The analysis of the test data used reveals that handoff costs are insignificant compared with that of link costs. Generally, there is also a substantial difference between the link cost of the least-costly switch and that of the second least-costly switch for each cell. Thus, the algorithm finds itself in presence of data that enables it to decide its assignment choices based almost exclusively on link cost (greatly simplifying its task) and to determine the order of selection of cells by taking into account costs difference between the two least-costly switches. Since the rules used to determine the order of selection and the assignment of cells are relatively efficient, these rules by themselves (without any ACO effects) provide solutions approaching the best results obtained by all methods developed. The effective application of heuristic parameters (authorizing a number of ants, cycles and loops greater than 1, and a partially random approach not solely based on selection and assignment rules) does not significantly improve the quality of solutions provided for the test data of André et al. However, in the double homing problem, where the handoff cost proportion is relatively important compared to link cost, and where the cost difference between the two least-costly switches is significantly less important, the ACO parameters will have an influence on the efficiency of execution and on the quality of the final solutions.

We first executed the ACO algorithm alone (unaided by a \(k\)-opt variant) and configured it to use the least-cost rule regarding assignment and to strictly enforce the switch capacity constraint, with all parameters set to a ‘null’ value (1 ant, 1 cycle and 1 loop, without a random factor and without pheromone deposits—i.e. the sole use of selection and assignment rules). In order to analyze the efficiency of effective application of parameters regarding single homing problem, we next executed this variant by having recourse to 7 ants to construct solutions, allowing 2 cycles without improvement for each of the 3 permitted loops, with a coefficient of 90% (corresponding to the \(q_0\) variable) for implementation of first argument (rather predictable, corresponding to the formula value

\[
Z_{\text{arg max}} \sum_{i=1}^{N_k} f(t_{ij}(t))/C_3(h_{ij})^b g
\]

regarding pseudo-random proportional rule (and of 10% (1\(−q_0\)) for implementation of second, random argument, corresponding to a random selection of a cell from all cells not yet assigned) concerning order of selection of cells, and with a coefficient of 100% regarding first argument of same pseudo-random rule (and of 0% for second argument) concerning assignment of cells. The parameters \(\alpha\) and \(\beta\) of the formula (value = \(\arg\max\sum_{i=1}^{N_k} k((\tau_{ijk}(t))^\alpha \star (\eta_{ijk})^\beta)\)) used to determine the order of selection.
of cells have respectively been set to 0.10 and 0.90 (the algorithm applying thus a factor of 0.10 to pheromone trails and of 0.90 to problem data—i.e. least-regret). The $\alpha$ and $\beta$ parameters for switch assignment have been set to 0 and 1 (the algorithm thus using solely problem data—i.e. least-cost). Finally, the right to deposit pheromones has been granted to all ants during the execution. Meanwhile, the ant obtaining the best solution of a cycle was granted the right to deposit double the normal amount of pheromones.

If one sums results of all 15 instances containing 200 cells and 7 switches, one observes a diminution of costs of only $-0.0068$ for the execution of algorithm with parameters set to the values indicated above, compared to the execution authorizing the use of only 1 ant, 1 cycle and 1 loop, and implementing no random factor and prohibiting the deposit of pheromones.

By assigning a relatively small value (10%) to the parameter $(1 - q_k)$ intended to incorporate a probabilistic feature (second argument of pseudo-random proportional rule, corresponding to a random selection of a cell from all cells not yet assigned), and a similarly small value to the factor ($\alpha$) applied to pheromone trails concerning the order of selection of cells, the algorithm uses the problem data most of the time in determining its choices, but integrates a random aspect for its search in order to diversify it. As we have mentioned above, this approach gives slightly improved results. The assignment of larger values to our variables unnecessarily increases execution time and often deteriorates quality of solutions. The increase of execution time can be explained by the diminished utilization of selection rules because of the more random aspect of the search process. Since the algorithm already performs its search in advantageous zones by applying selection rules, it frequently needlessly executes its search in more expensive regions when it has randomly been directed towards these zones. It needs time to converge again on zones containing the best solutions when it was initially directed towards these less favorable regions. By instead assigning a small value to the random elements of the heuristic, the algorithm generally follows its usual deterministic path (determined by problem data), but diversifies its search a little more, enabling it to encounter better solutions.

The inclusion of a random factor regarding assignment parameters produces less positive results since the deterministic rule consisting of assigning cells to their least-costly switch is relatively efficient in itself. Every time that the algorithm would assign a cell to a non-optimal switch by applying a random factor, the configuration costs of network would increase. Recall that cells are assigned to their switch in order of least-regret values. Thus, when the algorithm approaches the end of the assignment phase and that it is obliged to assign a cell to a non-optimal switch (because of exhaustion of capacities), it contains in its queue of not yet assigned cells, the ones having a less important difference between its two least-costly switches (because of use of the least-regret rule). Recall also that when a cell cannot be assigned to its optimal switch, it is assigned to the switch for which its costs are second (or third…) least important.

### 4.2. Hybridization of $k$-opt method with ACO heuristic

The results of ACO heuristic executed alone are similar (and sometimes very slightly better) to those obtained when executing the heuristic in combination with local $k$-opt search technique for small-size problems (e.g. 30 cells and 3 switches). This situation can be explained by the simplicity of these problems, which can be solved efficiently by only making use of the selection and assignment rules of ACO heuristic. However, the results for more complex large-size problems (e.g. 200 cells and 7 switches) are better when we combine the $k$-opt technique with the ACO heuristic.

For large-size problems, the algorithm providing best results when all ACO heuristic parameters are set to a null value (1 ant, 1 cycle and 1 loop, without any probabilistic factor and without any pheromone deposits—i.e. the sole use of selection and assignment rules) is the one applying the least-cost variant regarding assignment, authorizing temporary violation of capacities and executed in conjunction with second $k$-opt variant.

Amongst methods mentioned in the second section for comparison purposes, the algorithm having produced the best results is the one developed by André et al. [2] where the TS solution is provided to the CP heuristic. Fig. 2 compares this technique with our best variant. The results of the 15 instances comprising 200 cells and 7 switches are presented. Execution time for these two methods (with the same instances) is compared in Fig. 3. The sum of
the solution costs of 15 tests is 48,391 for the TS/CP algorithm. The sum of solutions provided by our best variant regarding same 15 tests amounts to 48,925, equivalent to a 1% difference compared to TS/CP heuristic. Meanwhile, execution time necessary to obtain all results is of 46.8 s for TS/CP algorithm, while it is of only 0.17 s when one does not take into account time necessary to read data on disk and of 4.13 s when one takes into account reading time for the ACO/k-opt heuristic. The CP heuristic executed alone provides, for its part, results amounting to 49,041 in 3.47 s. Results of our algorithm are 0.24% (i.e. 48,925) lower than this technique. TS executed with a solution provided by another method than CP generates in 95.44 s a solution 1.07% lower than ours, while same method used with ejection-chain technique produces a solution 1.04% lower than ours in 4.94 s. Finally, the TS heuristic developed by Houëto et al. [11,12,16] produces a solution that is 0.82% greater than ours in 42.43 s.

We obtain similar results for small-size (e.g. 30 cells and 3 switches) and medium-size (e.g. 100 cells and 5 switches) instances. We note that our heuristic is in general very slightly less efficient in finding good solutions for small-size instances. Relative to other algorithms, the efficiency of our heuristic improves as the size of the network increases.

Let us note that the execution time of our variant is between 1 and 2 hundredths of a second for every problem comprising 200 cells and 7 switches, if we assign a null value to all heuristic parameters and if we do not take into account reading of data on disk. The complexity of our algorithm when all the parameters are set to a null value is of $O(n^n n)$ (where $n$ equals number of cells), the algorithm using only 1 ant and executing only 1 cycle and 1 loop. Thus, the execution time of our algorithm is very steady and increases solely by a factor of $n^n n$ when the size of instances augments. It would therefore be possible to use the algorithm for instances comprising a larger amount of cells and switches without increasing execution time by a factor greater than $n^n n$. The quality of generated solutions would theoretically be the same and maybe even better, since the efficiency of our algorithm appears to improve when the size of instances becomes larger.

Finally, we have carried out an ‘extended’ execution of our variant by having recourse to 7 ants, authorizing the algorithm to execute 7 cycles without improvement for each of the 28 loops permitted, with a factor of 95% for the application of the first argument ($q_0$) variable, corresponding to the formula value $= \arg \max_{\tau(i)} k[(\tau(i))^\alpha * (\eta(i))^\beta]$ of the pseudo-random proportional rule (and of 5% (1-$q_0$) for the application of second argument, corresponding to a random selection of a cell from all cells not yet assigned) concerning order of selection of cells, and with a factor of 100% for the first argument of same rule (and of 0% for the second argument) concerning assignment of cells. Parameters $\alpha$ and $\beta$ regarding order of selection of cells have respectively been set to 0.10 and 0.90, while they were set to 0 and 1 regarding assignment. Each ant has been granted during this execution the right to deposit a quantity of pheromones, which was half the amount authorized to be deposited by the privileged ant obtaining the best solution of a cycle. The time needed for the algorithm to execute this alternative with the parameters set to values above was 4.5 s for the 15 tests containing 200 cells and 7 switches. The results of this execution are slightly better than those generated when the algorithm is executed with the parameters set to a null value (i.e. a difference of 0.6% instead of 1% in comparison to TS/CP algorithm). We have also authorized the algorithm to execute a number of cycles greater than 7. Meanwhile, we observe that the results produced are similar to previous ones and do not improve solutions.

### 4.3. Double homing problem

In order to avoid increasing the complexity of our double homing variant, our algorithm sequentially applies the procedure implemented for the single homing problem to the two data series corresponding to the two periods of the day, as previously defined, being careful not to count link costs twice when a cell is assigned to the same switch for both periods. It should be noted that it would be possible to increase the number of iterations being performed by algorithm if one wished to take into account more than two periods.

We observe that, in this problem, the parameters of our heuristic have a noteworthy influence on the generated results. Recall that the data (generated by André et al. [2]) used for this problem is different than that for the single homing problem. Handoff costs are significantly more important relative to link cost, in comparison with data used for our first problem. Furthermore, link costs for each switch (and for all cells) do not have an important difference between themselves, as opposed to instances of the single homing problem.

The results reveal that the use of the ACO heuristic alone provides better solutions than when combined with the local search $k$-opt technique. The effectiveness of the rules used by the ACO algorithm is thus confirmed by these results. When the algorithm has recourse to one of the $k$-opt variants in order to pursue its search, it uses new rules, which are obviously less efficient and consequently diminish the quality of solutions obtained with the ACO heuristic executed alone. Finally, results also indicate that application of the least-increase method regarding assignment generally provides better solutions than the least-cost technique.

The variant using the least-increase rule, respecting capacity constraints and executing the ACO heuristic alone (without any $k$-opt alternative) being the one that has generated the best results when all parameters were set to a null value, we have used it in order to execute our algorithm with different values for certain parameters. Results reveal that network configuration costs considerably diminish when one attributes a small value to probabilistic elements of algorithm (exclusively to parameters related to order of
selection of cells, the ones regarding assignment having the obligation to remain null during execution of the least-
increase alternative). These results confirm the efficiency of a partially random search regarding order of selection, enabling the algorithm to diversify its search towards new zones, which were not explored when using a purely deterministic approach based solely on initially known problem data. We have executed the algorithm using 4 ants, authorizing it to perform 3 cycles without any improvement for each of the 2 allowed loops, with factors of 90 and 10%, respectively, for the application of the first and second arguments of the pseudo-random proportional rule regarding order of selection of cells, and with factors of 100 and 0%, respectively, for the first and second arguments of same rule regarding assignments of cells. The \( \alpha \) and \( \beta \) parameters regarding order of selection of cells were respectively set to 0.10 and 0.90, while they were set to 0 and 1 regarding assignment. Each ant was granted during the execution the right to deposit a quantity of pheromones. Meanwhile, the privileged ant obtaining the best solution during a cycle was granted the right to deposit twice as much pheromone than that of other ants. The execution of this algorithm with its parameters set to the above values has generated results providing a 13.8% diminution compared to results obtained with execution of algorithm with all of its parameters set to a null value.

5. Conclusion

We have applied the relatively recent ACO meta-heuristic to the single and double homing assignment problems of cells to switches in a mobile network. We have also combined execution of the heuristic with the local \( k \)-opt optimization technique. Several variants of the methods have been created.

In comparison with researches previously done regarding the single homing problem, one can conclude that our methods are relatively efficient regarding both quality of solutions provided and execution time. Solutions generated by our best variant are only 1% more costly than those of methods having given best solutions, but are obtained in a shorter execution time. Recall that the complexity of our analyzed variant is in the order of \( O(n^3) \) when ACO heuristic parameters are set to a null value. It would thus be possible to increase the size of problem instances without augmenting the algorithm’s execution time by a factor larger than \( n^2 \) and obtain solutions with an equivalent quality. Let us note that the effectiveness of our algorithm appears to increase with the size of instances relative to other methods mentioned in this paper. It would be interesting to apply it to instances with a larger amount of cells and switches in order to verify if this trend continues.

Regarding the double homing problem, we have performed a simplified implementation. The cost structure of data used for tests related to this problem is different than one related to data used for the single homing problem, inciting us more to adjust the value of the ACO heuristic parameters relative to the single homing problem in order to increase quality of solutions. Our algorithm enables us meanwhile to obtain economic solutions with instances containing a cost structure similar to one found in files used for tests regarding the single homing problem (by having solely recourse to rules of selection for determining order of selection and assignment), and with instances having a cost structure analogous to one of files used for the double homing variant.

However, our algorithm somewhat lacks robustness with respect to parameter values. Certain values significantly increase algorithm execution time without necessarily generating better results (increase in the number of cycles and loops for instance). Also, values of some parameters appreciably deteriorate quality of solutions provided (setting for instance a positive value to the parameter concerning the second argument—random component—of the pseudo-random proportional rule for cell assignments). This phenomenon can be explained by the effectiveness of the rules implemented regarding assignment (least-cost or least-increase), which are partly ignored when using a random approach. The appropriateness of using such a random tactic regarding assignment then dwindles. It would nevertheless be interesting to substitute a new non-random rule to the second argument of the pseudo-random proportional rule in order to determine if it is possible to obtain better results with such a substitution.

Our implementation for the double homing problem solely performs a static assignment regarding the physical links to put up between each cell and the switches of the network. The algorithm could be used to determine the best network configuration according to the number of known or anticipated calls for each cell, for each considered period of the day. It would afterwards be possible to design a dynamic algorithm that would take into account the existing physical links, which would determine in real time the appropriate moment for changing assignment (in respect of their management) of cells having two links, thus reducing the total number of handoffs occurring in the network.

References