A GREEDY RANDOMIZED ADAPTIVE SEARCH PROCEDURE FOR THE POINT-FEATURE CARTOGRAPHIC LABEL PLACEMENT

Gildásio Lecchi Cravo[1], Glaydston Mattos Ribeiro[2,3] and Luiz Antonio Nogueira Lorena[3*]

[1]UNIARACRUZ – Faculty of Aracruz

29.194-910, Aracruz – ES – Brazil. Tel/FAX: +55 27 3256-1102

E-mail: lecchi@fsjb.edu.br

[2]UFES – Federal University of Espírito Santo

29.933-480, São Mateus – ES – Brazil. Tel/FAX +55 27 3763-6273

E-mail: glaydstonribeiro@ceunes.ufes.br

[3]INPE – Brazilian Space Research Institute

12.227-010, São José dos Campos – SP – Brazil. Tel/FAX: +55 12 3945-6555/3945-6357

E-mail: {glaydston, lorena}@lac.inpe.br

Abstract

The point-feature cartographic label placement problem (PFCLP) is an NP-hard problem which appears during the production of maps. The labels must be placed in predefined places avoiding overlaps and considering cartographic preferences. Due to its high complexity several heuristics have been presented searching for approximated solutions. This paper proposes a greedy randomized adaptive search procedure (GRASP) for the PFCLP that is based on its associated conflict graph. The computational results show that this metaheuristic is a good strategy for PFCLP, generating better solutions than all those reported in the literature in reasonable computational times.

Keywords: GRASP, Heuristic, Label placement.

[*] Corresponding author

1. Introduction

The cartographic label placement problem is an important task in automated cartography and Geographical Information Systems (GIS). Labels convey information about objects (or features) in graphical displays like graphs, networks, diagrams, or cartographic maps (Wolff, 1999). Each feature that needs to be labeled has a number of positions where its label can be placed. However, it is essential that all labels must be placed without overlaps.

This paper is concerned with the placement of labels for point features. The point-feature cartographic label placement problem (PFCLP) is the problem of placing text labels to point features on a map, graph or diagram in such a manner so as to maximize the legibility of the picture. Figure 1 shows an example of the PFCLP. Note that there are several obscured areas and consequently we cannot read some names.

Figure 1 – An example of a map with some overlapping labels (see arrows) (Ribeiro and Lorena, 2006)

For each point there exists a set of possible label positions also known as a set of candidate positions and the objective is to find a combination of these candidate positions that produces the best legibility in a map. This set of potential label positions indicates their desirability also known as cartographic standardization (Christensen et al., 1995). Figure 2 shows a group of 8 candidate positions for a point, where the numbers indicate the cartographic preferences. Position number 1 is the most preferred.

Figure 2 – Set of eight candidate positions for a point (Christensen et al., 1995)

The PFCLP is an optimization problem shown to be NP-Hard (Formann and Wagner, 1991; Marks and Shieber, 1991). Thus, exact solution techniques are not usual and several heuristics and metaheuristics have been proposed. Some approaches consider a conflict graph generated by the problem presenting the following structure where candidate positions are associated to vertices and the potential overlaps (or conflicts) are associated to edges.

Let N be the number of points that must be labeled and P the number of candidate positions for each point. $G=\{V,E\}$ is the corresponding conflict graph where $V=\{v_1, v_2, \ldots, v_{N*P}\}$ is the set of candidate positions and $E=\{(v_i, v_j):i,j\in V, i\neq j\}$ the potential conflicts between candidate positions.

Figure 3. Conflict graph for PFCLP. (a) Problem, (b) Conflict graph and (c) Optimal solution (Ribeiro and Lorena, 2006).

Figure 3(b) shows the conflict graph generated from problem shown in Figure 3(a), and Figure 3(c) shows the optimal solution for this problem. The proportion of conflict free labels assesses the quality of labeling (Ribeiro and Lorena, 2005; Yamamoto et al, 2002; Christensen et al, 1995). So, considering the problem shown in Figure 3, the solution provided in Figure 3(c) has 100% of conflict free labels.

In this paper we consider a greedy randomized adaptive search procedure (GRASP) applied to the conflict graph associated to the PFCLP. GRASP is a metaheuristic for finding approximate solutions to combinatorial optimization problems (Resende and Ribeiro, 2003). It was first introduced by Feo and Resende (1989) in a paper describing a probabilistic heuristic for set covering problems.

GRASP is an iterative method that has two distinct phases. The first one is a constructive phase that blends greedy and random construction either by using greediness to build a restricted candidate list (RCL) and using randomness to select an element from the list, or by using randomness to build the list and using greediness for the selection (Resende and Ribeiro, 2003). The second phase is represented by a local search algorithm, exploring a neighborhood of the current solution provided by first phase. The solution resulting from the local search becomes the new best solution if the current solution is improved.

The GRASP implementation for PFCLP is tested upon instances proposed in the literature and the computational results were successful compared with the best ones reported.

The paper is organized as follows. Next section presents a review about the PFCLP, followed by Section 3 that presents a brief review of the GRASP and describes the GRASP proposed for the PFCLP. Section 4 presents the computational results and Section 5 concludes with some future research directions.

2. Literature Review

This paper deals with label placement of point features but the literature reports feature problems involving lines and areas such as roads and states. See the map labeling bibliography (Wolff and Strijk, 2005) for further details. The typical input of the PFCLP is a discrete set of candidate positions for the points (see Figure 2).

There are three different approaches for PFCLP in the literature. The first one searches the largest number of conflict free labels, even not labeling all points. This problem can be seen as a

Maximum Independent Set Problem (MISP) (Zoraster, 1990; Strijk et al., 2000). The second approach also searches the maximum number of conflict free labels, however in this case, all points must be labeled. This problem is known as Maximum Number of Conflict Free Labels Problem (MNCFLP) (Ribeiro and Lorena, 2005). This approach has been used by several researchers such as Christensen et al (1994; 1995), Verner et al (1997) and Yamamoto and Lorena (2005). The last approach, first introduced by Ribeiro and Lorena (2005; 2006), the Minimum Number of Conflicts Problem (MNCP) is considered, where all points must be labeled whereas the number of conflicts is minimized.

On the other hand, there are some more general models, which allow the labels to move around their features. They are known as slider models (Klau and Mutzel, 2000; 2003). However, the model we consider does not take sliding into account.

If the PFCLP is considered as a MISP, substantial research has been done in the literature, and different algorithms and techniques are presented. Zoraster (1986, 1990 and 1991) formulated mathematically the PFCLP with conflict constraints and if the points could not be labeled they are related to dummy candidate positions of high cost. Zoraster used a Lagrangean relaxation and obtained some computational results for small-scale data sets. Moreover, Zoraster (1997) used a Simulated Annealing algorithm for solving PFCLPs in petroleum industry.

Strijk et al. (2000) proposed other mathematical formulations exploring some cut constraints. These cuts are based on cliques and appeared before in the works of Moon and Chaudhry (1984) and Murray and Church (1996). They applied and proposed several heuristics: Simulated Annealing, Diversified Neighborhood Search, *k-opt* and Tabu Search. The last one showed the better results for their instances.

The Maximum Number of Conflict Free Labels Problem (MNCFLP) was examined in several papers. Hirsch (1982) developed a Dynamic Algorithm of label repulsion, where labels in conflicts are moved trying to avoid a conflict. The algorithm defines repelling forces for overlapping labels and computes translation vectors for them. After translation, this process is repeated and hopefully, a labeling with few overlaps appears after a number of iterations. Christensen et al. (1994; 1995) proposed an Exhaustive Search Approach, alternating positions of the labels that were previously positioned. Christensen et al. (1995) also proposed a Greedy Algorithm and a Discrete Gradient Descent Algorithm but these algorithms have difficulty of escaping from local maximum. Verner et al. (1997) applied a Genetic Algorithm with mask such that if a label is in conflict the changing of positions are allowed by crossover operators. Yamamoto et al. (2002) proposed a Tabu Search algorithm that provides good results when compared with the literature. Schreyer and Raidl (2002) applied Ant Colony System but the results were not interesting when compared to the ones obtained by Yamamoto et al. (2002). Yamamoto and Lorena (2005) developed an exact algorithm for small instances of PFCLP and applied the Constructive Genetic Algorithm (CGA) proposed by Lorena and Furtado (2001), to a set of large-scale instances. The exact algorithm was applied to instances of 25 points and the CGA was applied to instances up to 1000 points. Recently, Yamamoto et al (2005) proposed an algorithm called by FALP that provided better results than CGA using the same set of instances proposed by Yamamoto et al. (2002).

The PFCLP considered as a MISP or MNCFLP can generate large conflict graphs that become hard to deal with. Wagner et al. (2001) presented an approach to reduce the conflict graph provided by a PFCLP. They proposed three rules to reduce the size of the conflict graph without altering the set of optimal solutions. Moreover, they combined these rules with a heuristic yielding near-optimal solutions to a set of instances.

Considering now the PFCLP as a MNCP, Ribeiro and Lorena (2005; 2006) have proposed two 0-1 integer linear programming models and a Lagrangean heuristic that have presented the best solutions in the literature for the instances proposed by Yamomoto et al (2002). The second formulation proposed by the authors has a compact number of constraints and is reproduced bellow.

$$v(MNCP) = Min \sum_{i=1}^{N}\sum_{j=1}^{P_i}\left(w_{i,j}x_{i,j} + \sum_{c \in C_{i,j}} y_{i,j,c} \right) \quad (1)$$

Subject to:

$$\sum_{j=1}^{P_i} x_{i,j} = 1 \quad \forall\, i = 1 \ldots N \quad (2)$$

$$\left|C_{i,j}\right| x_{i,j} + \sum_{(k,t)\in S_{i,j}} x_{k,t} - \sum_{c \in C_{i,j}} y_{i,j,c} \le \left|C_{i,j}\right| \quad \forall\, i = 1 \ldots N, \ \forall\, j = 1 \ldots P_i \quad (3)$$

$$x_{i,j}, x_{k,t} \text{ and } y_{i,j,c} \in \{0,1\} \quad \forall\, i = 1 \ldots N, \ \forall\, j = 1 \ldots P_i, \ c \in C_{i,j} \quad (4)$$

Where:

- N is the number of points to be labeled and P_i is the set of candidate positions of point i;
- $x_{i,j}$ is a binary variable such as $i \in N$ and $j \in P_i$;
- $w_{i,j}$ is the cartographic preference assigned to each candidate position. It allowed us to prioritize some candidate positions as shown in Figure 2;
- $S_{i,j}$ is a set of index pairs $(k,t){:}k>i$ of candidate positions such that $x_{k,t}$ has potential conflict with $x_{i,j}$;

- $C_{i,j}$ is a set with all points that contain candidate positions in conflict with the candidate position $x_{i,j}$; and
- $y_{i,j,c}$ is a conflict variable between the candidate position $x_{i,j}$ and the point $c \in C_{i,j}{:}c>i$.

Constraint (2) ensures that each point must be labeled with one candidate position. Constraint (3) ensures that if vertices with potential conflicts are chosen to compose the solution, the object function described in Equation (1) will be penalized. And Equation (4) indicates that all variables in the model are binaries.

3. GRASP for the PFCLP

The GRASP (greedy randomized adaptive search procedure) is a multistart iterative process of two phases: a constructive phase, in which a feasible solution is produced, and local search phase, in which a local optimum in the neighborhood of the solution is sought (Feo and Resende, 1995). These phases are repeated considering a maximum number of iterations or an alternative criterion to stop the iterative process. Figure 4 shows the GRASP pseudo-code for a minimization problem. The value of the best solution is stored in f^* and i_{max} GRASP iterations are executed.

Figure 4 – A basic GRASP for minimization problem (Resende and Ribeiro, 2005).

Considering the pseudo-code described in Figure 4, the procedures *GreedyRandomizedConstruction* and *LocalSearch* must be defined using the available information about the problem to be solved. The feasible solution is iteratively constructed in the first phase, one element at a time. The choice of the next element to be added is determined by ordering all candidate elements in a candidate list with respect to its contribution to the objective function. The list of best candidates is called Restricted Candidate List (RCL).

There are several practical applications using GRASP. Bard and Feo (1989) described a method for efficiently sequencing the cutting operations associated with manufacture of discrete parts. The authors use GRASP to provide a lower bound in a lagrangean relaxation of the problem. Feo et al (1991) used GRASP for a difficult single machine scheduling problem. Laguna et al (1991) combined GRASP and Tabu Search for solving just-in-time scheduling in parallel machines. Resende and Resende (1997) described a GRASP for routing permanent virtual circuits (PVC) for frame relay in telecommunications systems. Their objective was to minimize PVC delays while balancing trunk loads. GRASP was also applied in a complex vehicle-scheduling problem with tight time windows and with additional constraints providing interesting results (Atkinson, 1998).

GRASP has been also applied in theoretical applications. Feo et al (1994) applied it for approximately solving the maximum independent set problem obtaining good results. Resende and Ribeiro (1997) proposed a GRASP for graph planarization. Abello et al (1999) applied a GRASP for solving the maximum clique problem and maximum quasi-clique problem. They discuss some graph decomposition schemes that breaks up the original problem into several pieces of manageable dimensions. In this work, the construction phase uses vertex degrees as a guide for construction and a change heuristic for the local search.

There are also GRASP enhancements in literature. The reactive GRASP, proposed by Prais and Ribeiro (2000), is a GRASP that does not use a fixed value for the basic parameter defining the RCL length during the constructive phase. Reactive GRASP self-adjusts the RCL according to the quality of the solutions previously found. Laguna and Martí (1999) incorporated to GRASP a path relinking strategy proposed by Glover (1996) searching for improved outcomes. Recently, Resende and Ribeiro (2005) presented several advances and applications for the GRASP with path relinking. For a good review about GRASP and its applications, see Festa and Resende (2002), Aiex et al (2003) and Resende and Ribeiro (2003).

The GRASP for the PFCLP proposed in this paper deals with the MNCP approach proposed by Ribeiro and Lorena (2005) minimizing the number of conflicts in the corresponding conflict graph.

The constructive phase:

Considering the PFCLP represented by a conflict graph, the constructive heuristic is based in the vertex degrees, as done in some works of the literature (Abello et al., 1999; Feo et al., 1994). The degree of a vertex is the number of incident edges in that vertex so it represents a measure of labels in conflict.

Let $G=(V,E)$ be a conflict graph as described in Section 1 where V is a set of vertices (candidate positions) and E is a set of edges (conflicts between vertices), and let *RCL_Length* be the size of the restricted candidate list (RCL). Thus, the pseudo-code of the constructive phase can be written as shown in Figure 5.

Figure 5 – A constructive heuristic for the PFCLP.

The method *CreateRCL* is responsible to create the restricted candidate list. First, considering the current conflict graph, it computes a weight for each vertex based on degrees, cartographic preferences and potential conflicts with the current solution *Sol*, and sorts the vertices in an increase order. After that, only the first *RCL_Length* are considered as restricted. This list is created using the current graph because the graph is reduced at each iteration of the procedure shown in Figure 5. When the procedure selects an element to enter in the current solution, actually this element is a candidate position of some point *i* and a method to reduce the current graph must be applied. This method, called *ReduceActiveVertices*, receives the list of current vertices represented by *V_Greedy* and the selected candidate position *v*, and removes from *V_Greedy* all candidate positions of the point *i* that has *v* as a candidate position.

When the graph is reduced, the remaining graph still presents some vertices that are in potential conflict with the current solution *Sol*. Thus, to order the candidate position and to select the restricted list, the method *CreateRCL* calculates the weight of each vertex considering the following equation:

$$Weight(x) = CartographicPreference(x) + Degree(x) + M * PotentialConflicts(Sol,x) \quad (5)$$

Where:

- *CartographicPreference(x)* returns the cartographic preference of the vertex *x*;
- *Degree(x)* returns the degree of the vertex *x* considering the current conflict graph;
- *PotentialConflicts(Sol, x)* returns the number of the potential conflicts between *x* and the current solution *Sol*; and
- *M* is a coefficient used to penalize the vertex *x* if it presents potential conflicts with the current solution *Sol*.

The function represented by Equation (5) ensures that if a vertex *x* has high degree and several potential conflicts with the current solution *Sol*, it will appear at the end of the list when the vertices are sorted, avoiding conflicts.

Figure 6 shows two iterations of a theoretical example where all cartographic preferences are equal to one. For this example, *RCL_Length = 4* and *M =10* in Equation (5). Note that at first iteration represented in Figure 6(a), the randomized selected vertex was v_1 and consequently all allowed candidate positions of the point are removed from the conflict graph. In second iteration (see Figure 6(b)), the conflict graph is reduced and all vertex degrees are recalculated. Besides, the vertex v_{13} has a potential conflict with the vertex v_1 selected at iteration one. Thus, this vertex is penalized appearing at the end of sorted list, consequently reducing its chances of being selected.

Figure 6 – An example of the constructive heuristic for the PFCLP.

This process is repeated until a feasible solution is provided. If the considered RCL length in some iteration is greater than the remaining vertices in the reduced graph, the algorithm must consider the total of these remaining vertices as RCL length.

The local search phase:

The second phase of the GRASP for the PFCLP uses a local search heuristic that tries to change a candidate position assigned for each point *i* in the constructive heuristic to another valid position $j \in P_i$, searching for a solution that reduces the objective function described in Equation (1). The proposed local search heuristic is shown in Figure 7.

The procedure shown in Figure 7 can be resumed as follows: Line 11 tests for each point another valid candidate position. Lines 9 to 18 perform the storage of the best change and after check all possible changes, the best one is performed (see lines 19 and 20). While the algorithm finds a new solution, the search is repeated in the neighborhood of this new solution (line 5). Take a feasible solution tested with the same objective function of the best solution found (line 12) if this new solution increases the number of conflict free labels.

Figure 7 – Local search heuristic.

The proposed GRASP is very simple to implement but it is dependent on the number of iterations and the size of restricted candidate list. Thus, to define the best values for these parameters, it was applied over standard sets of randomly generated points proposed by Yamamoto et al (2002) available at http://www.lac.inpe.br/~lorena/instancias.html. The computational results presented in next section show that the GRASP produced better results than those reported in the literature.

4. Computational Results

The GRASP was coded in C++ running on a Pentium IV 2.80 GHz processor and 512 MB of RAM memory. The instances proposed by Yamamoto and Lorena (2002) are composed of twenty five instances for each value for the number of points *N*. As done by Zoraster (1990), Yamamoto et al (2002), Yamamoto and Lorena (2005) and Ribeiro and Lorena (2005), for all problems the cartographic preferences were not considered, i. e., the preferences are set to 1 for all the candidate positions, being the number of those positions equal to 4. It thus allowed comparing of the GRASP results to the ones present in the literature.

After a number of initial experiments the maximum number of iterations (i_{max}) was fixed in 100. With more iterations the results did not improved as expected. The size of the restricted candidate list was set in a range of 2 to 10, and the results are influenced significantly for each value in this range. Also, for greater RCL the solutions are not significantly improved.

Tables 1 to 4 report the main results found varying the RCL length and performing the GRASP 10 times. In these tables, the first column indicates the number of points to be labeled and the next three columns are average results over 10 runs and these averages are calculated over the average results for twenty five instances for each number of points. Thus, the second column shows the average results over 10 runs for the number of remaining edges, followed by average results of the number of conflict labels and of the percentages of conflict free labels that is calculated as $\left(\frac{N - Number\ of\ Conflict\ Labels}{N}\right)*100$. The fifth through seventh column present the best average solution provided by the GRASP among 10 runs.

Note that the better results were found when considering 5 and 6 for the RCL length. However the computational times are not satisfactory. Considering only Tables 2 and 3, the computational times for the most difficult problems (with 750 and 1000 points) are varying from 40 to 112 seconds. So, the GRASP approximately elapsed 2 minutes for solving the largest instances and for some applications, this time can be considered high.

Table 1. Average results obtained with GRASP for *RCL_Length = 2*.

Table 2. Average results obtained with GRASP for *RCL_Length = 5*.

Table 3. Average results obtained with GRASP for *RCL_Length = 6.*

Table 4. Average results obtained with GRASP for *RCL_Length = 10*.

Trying to decrease these computational times, the reducing technique proposed by Wagner et al (2001) was applied to the conflict graphs before starting our experiments. For all instances with 100 and 250 points the technique produced considerable reductions but the performance did not repeated for the instances with 25, 750 and 1000 points. The GRASP results are shown in Tables 5 to 8. These tables have the same columns presented in Table 1.

Table 5. Average results obtained with GRASP for *RCL_Length = 2* using the reduction proposed by Wagner et al (2001).

Table 6. Average results obtained with GRASP for *RCL_Length = 5* using the reduction proposed by Wagner et al (2001).

Table 7. Average results obtained with GRASP for *RCL_Length = 6* using the reduction proposed by Wagner et al (2001).

Table 8. Average results obtained with GRASP for *RCL_Length = 10* using the reduction proposed by Wagner et al (2001).

Again the best results are found for RCL length 5 or 6. Overall, the quality of the solutions are improved and the computational times are reduced with the conflict graph reduction. Considering

Tables 6 and 7, for instances with 750 and 1000 points, the computational times varied from 5 to 26 seconds against 40 and 112 without the reduction, respectively.

Considering Tables 5 to 8 and the columns showing the results for average conflict free labels over 10 GRASP runs, for *RCL_Length = 5*, the GRASP produced better results for the instances with 750 points and for *RCL_Length = 6*, it improved the instances with 25 and 1000 points. For others values for the RCL length the results are close to the best ones found.

Figure 8 summarizes the average results shown at Tables 5 to 8. Note that the squares and triangles indicate that, respectively, the solutions provided by GRASP when RCL length are 5 and 6 outperformed other results.

Figure 8 – Comparison among proportion of conflict free label considering the average results over 10 GRASP runs and the reduction proposed by Wagner et al (2001).

Now the best results found in this paper are compared to the best ones of the literature described in the works of Yamamoto and Lorena (2005) and Ribeiro and Lorena (2006). Table 9 reports the best average percentages of conflict free labels found using the GRASP proposed and the best results found in the literature. For this comparison, we used the results found with the reduced graphs and with RCL length defined as 5 and 6. Note that the approaches have different objectives, however the GRASP found better results to PFCLP than all those reported in the literature. The computational times are not compared since the computational tests were performed in different machines.

Table 9. Comparison with the literature.

5. Conclusions

This paper has presented a greedy randomized adaptive search procedure (GRASP) for the point-feature cartographic label placement problem. GRASP is a metaheuristic that has been applied in several practical and theoretical applications.

Considering an optimization point of view, the proposed GRASP presented better results than all reported in the literature using a reduction technique for conflict graphs. The GRASP results were better than well-known metaheuristics such as Simulated Annealing, Tabu Search and Genetic Algorithm. Besides, it improved the best-known solutions in the literature that were provided by a lagrangean relaxation.

The research can be continued combining GRASP and path relinking techniques, during the GRASP iterations or in a post-optimization phase. However it is necessary to study the application of path relinking in the pool of elite solutions without significant increase in computational times.

Acknowledgements

The authors acknowledge the useful comments and suggestions of two anonymous referees and Ribeiro and Lorena acknowledge Conselho Nacional de Desenvolvimento Científico – CNPq for partial financial support.

References

1. Abello, J., Pardalos, P.M., Resende, M.G.C. On Maximum Clique Problems in Very large graphs. *In*: Abello, J., Vitter, J. (Eds.). External memory algorithms and visualization, volume 50 of DIMACS Series on Discrete Mathematics and Theoretical Computer Science, 1999.

2. Aiex, R. M., Binato, S., Resende, M. G. C. Parallel GRASP with path-relinking for job shop scheduling. Parallel Computing 2003; 29:393-430.

3. Atkinson, J.B. A greedy randomized search heuristic for time-constrained vehicle scheduling and the incorporation of a learning strategy. Journal of the Operational Research Society 1998; 49:700-708.

4. Bard, J.F., Feo, T.A. Operations Sequencing in Discrete Parts Manufacturing. Management Science 1989; 35:249-255.

5. Christensen J., Marks J., Shieber S. Placing Text Labels on Maps and Diagrams. In: Heckbert P. (Ed.). Graphics Gems IV. Cambridge: Academic Press, 1994.

6. Christensen J., Marks J., Shieber S. An Empirical Study of Algorithms for Point-Feature Label Placement. ACM Transactions on Graphics 1995; 14(3): 203-232.

7. Feo, T. A., Resende, M. G. C. A Probabilistic Heuristic for a computationally difficult set covering problem. Operations Research Letters 1989; 8:67-71

8. Feo, T.A., Venkatraman, K., Bard, J.F. A GRASP for a difficult single machine scheduling problem. Computers and Operations Research 1991; 18:635-643.

9. Feo, T.A., Resende, M.G.C., Smith, S.H. A Greedy Randomized Adaptive Search Procedure for Maximum Independent Set. Operations Research 1994; 42:860-878.

10. Feo, T.A., Resende, M.G.C. Greedy Randomized Adaptive Search Procedures. Journal of Global Optimization 1995; 6:109-133.

11. Festa, P. Resende, M. G. C. GRASP: An Annotated bibliography. *In:* Ribeiro, C. C. and Hansen, P. (Eds.). Essays and Surveys on Metaheuristics. Kluwer Acadmic Publishers, 2002. p. 325-367.

12. Formann, M. and Wagner, F. A packing problem with applications to lettering of maps. *In:* Proceedings of the Seventh Annual Symposium on Computational Geometry, 1991; 281-288. North Conway, New Hampshire, July. ACM.

13. Glover, F. Tabu Search and Adaptive Memory Programming – Advances, applications and challenges. *In*: Barr, R. S., Helgason, R. V. and Kennington, J.L. (Eds.). Interfaces in Computer Science and Operations Research. Kluwer, 1996. p. 1-75.

14. Hirsch S. A. An Algorithm for Automatic name Placement around Point Data. American Cartographer 1982; 9(1): 5-17.

15. Klau, G. W. and Mutzel, P. Optimal labeling of point features in the slider model. *In*: D.-Z. Du, P. Eades, V. Estivill-Castro, X. Lin, and A. Sharma, editors, Proc. 6th Annual International Computing and Combinatorics Conf. (COCOON'00), volume 1858 of Lecture Notes in Computer Science 2000, Springer-Verlag. p. 340-350.

16. Klau, G. W. and Mutzel, P. Optimal labeling of point features in rectangular labeling models. Mathematical Programming, Ser. B, 2003; 94:435-458.

17. Laguna M., Gonzalez-Velarde, J.L. A search heuristic for just-in-time scheduling in parallel machines. Journal of Intelligent Manufacturing 1991; 2:253-260.

18. Laguna, M., Martí, R. GRASP and Path Relinking for 2-layer straight line crossing minimization. INFORMS Journal on Computing 1999; 11:44-52.

19. Lorena L. A. N., Furtado, J. C. Constructive genetic algorithm for clustering problems. Evolutionary Computation 2001; 9(3): 309-327.

20. Marks, J. and Shieber, S. The computational complexity of cartographic label placement. Technical Report TR-05-91, 1991. Harvard University, March.

21. Moon I. D., Chaudhry S. An analysis of network location problems with distance constraints. Management Science 1984; 30: 290-307.

22. Murray A. T., Church R. L. Solving the anti-covering location problem using lagrangian relaxation. Computers and Operations Research 1996; 24(2): 127-140.

23. Prais M., Ribeiro, C.C. Reactive GRASP: An application to a matrix decomposition problem in TDMA traffic assignment. INFORMS Journal on Computing 2000; 12:164-176.

24. Resende, L.I.P., Resende, M.G.C. A GRASP for frame relay PVC routing. Technical report, AT&T Labs Research, Florham Park, NJ 07932 USA, 1997.

25. Resende, M.G.C., Ribeiro, C.C. A GRASP for graph planarization. Networks 1997; 9:173-189.

26. Resende, M. G. C., Ribeiro, C. C. Greedy randomized adaptive search procedures. In: Glover, F. and Kochenberger, G. (Eds.). Handbook of Metaheuristics. Kluwer Academic Publishers, 2003. p. 219-249.

27. Resende, M. G. C., Ribeiro, C. C. GRASP with path-relinking: recent advances and applications. In: Ibaraki, T., Nonobe, K. and Yagiura, M. (Eds.). Metaheuristics: Progress as Real Problem Solvers. Kluwer Academic Publishers, 2005. p. 29-63.

28. Ribeiro G. M., Lorena, L. A. N. Heuristics for cartographic label placement problems. Computers and GeoSciences 2005; 32(6):739-748.

29. Ribeiro G. M., Lorena, L. A. N. Lagrangean relaxation with clusters for point-feature cartographic label placement problems. Computers and Operations Research, 2006. To Appear.

30. Schreyer M., Raidl G. R. Letting ants labeling point features. In Proc. of the 2002 IEEE Congress on Evolutionary Computation at the IEEE World Congress on Computational Intelligence; 2002, 1564-1569.

31. Strijk T., Verweij B., Aardal K. Algorithms for maximum independent set applied to map labeling. Available at ftp://ftp.cs.uu.nl/pub/RUU/CStechreps/CS-2000/2000-22.ps.gz., 2000.

32. Verner O. V., Wainwright R. L., Schoenefeld D. A. Placing text labels on maps and diagrams Using Genetic Algorithms with Masking. INFORMS Journal on Computing 1997; 9: 266-275.

33. Yamamoto M., Camara G., Lorena L.A.N. Tabu search heuristic for point-feature cartographic label placement. GeoInformatica and International Journal on Advances of Computer Science for Geographic Information Systems. Kluwer Academic Publisher. Netherlands, 2002, 6(1): 77-90.

34. Yamamoto M., Lorena L.A.N. A constructive genetic approach to point-feature cartographic label placement. In: Ibaraki, T., Nonobe, K. and Yagiura, M. (Eds.). Metaheuristics: Progress as Real Problem Solvers. Kluwer Academic Publishers, 2005. p. 285-300.

35. Yamamoto, M., Camara, G., Lorena, L. A. N. Fast point-feature label placement for real time screen maps. GEOINFO 2005 - VII Brazilian Symposium on GeoInformatics - Campos do Jordão – São Paulo, Brazil. Available at http://www.lac.inpe.br/~lorena/missae/sbc_Missae.pdf.

36. Wagner, F., Wolff, A., Kapoor, V., Strijk, T. Three rules suffice for good label placement. Algorithmica 2001; (30): 334-349.

37. Wolff, A. Automated label placement in theory and practice. PhD thesis, Fachbereich Mathematik und Informatik, Freie Universität Berlin, 1999. pp. 153.

38. Wolff, A., Strijk, T., (2005). The map labeling bibliography. URL on June 6th: http://i11www.ilkd.uni-karlsruhe.de/~awolff/map-labeling/bibliography/

39. Zoraster S. Integer programming applied to the map label placement problem. Cartographica 1986; 23(3): 16-27.

40. Zoraster S. The solution of large 0-1 integer programming problems encountered in automated cartography. Operations Research 1990; 38(5): 752-759.

41. Zoraster S. Expert systems and the map label placement problem. Cartographica 1991; 28(1): 1-9.

42. Zoraster S. Practical results using simulated annealing for point feature label placement. Cartography and GIS, 1997; 24(4): 228-238.

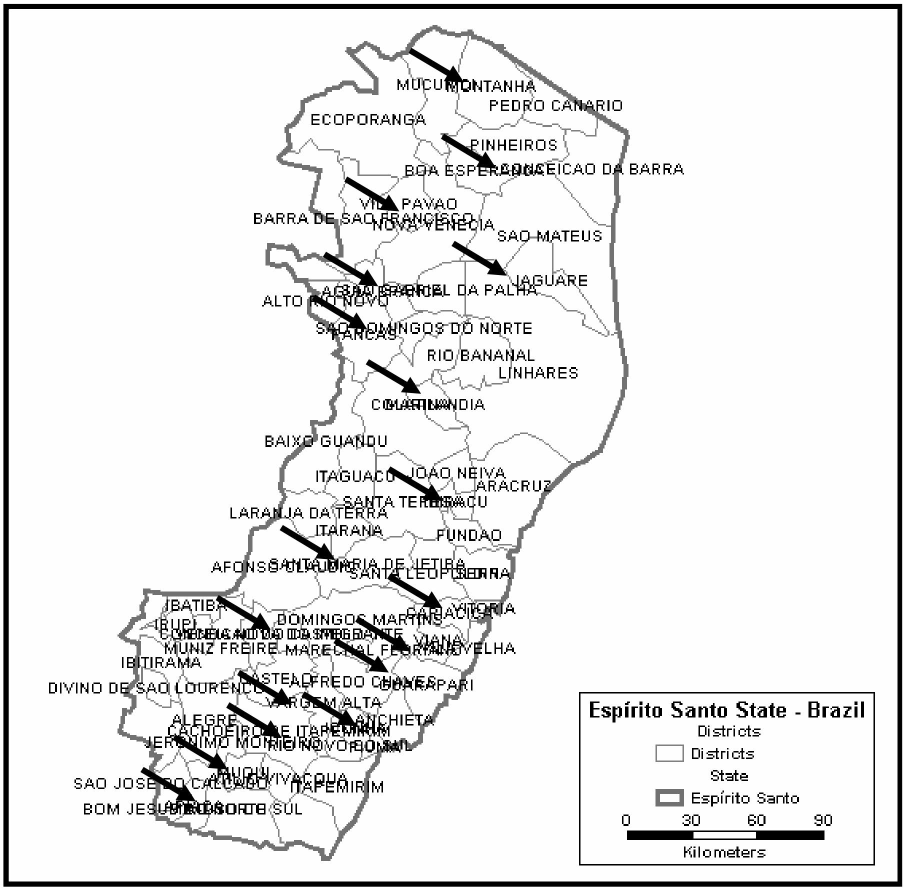


Figure 1 – An example of a map with some overlapping labels (see arrows) (Ribeiro and Lorena, 2006)

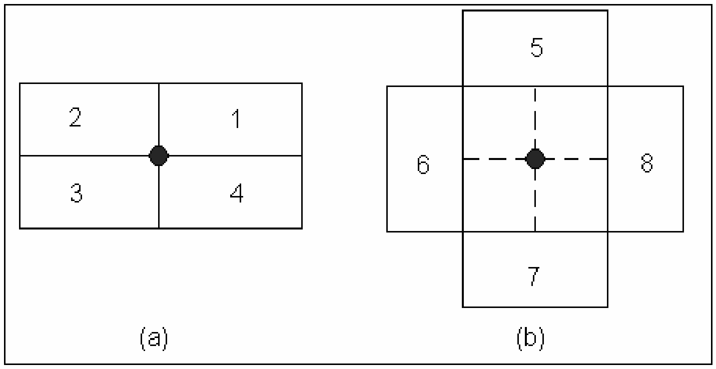


Figure 2 – Set of eight candidate positions for a point (Christensen et al., 1995)

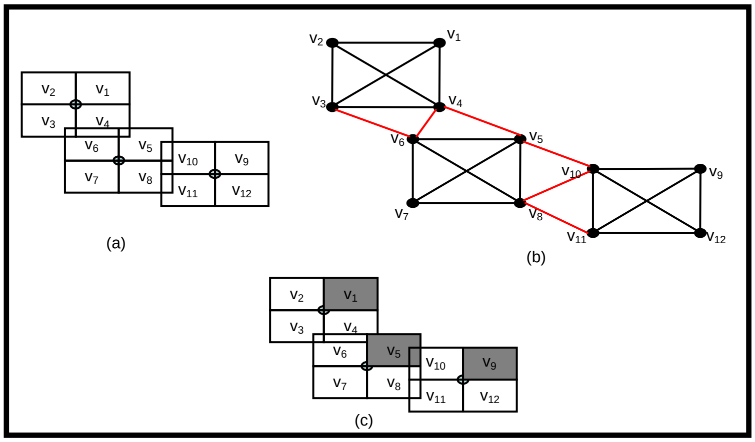


Figure 3. Conflict graph for PFCLP. (a) Problem, (b) Conflict graph and (c) Optimal solution (Ribeiro and Lorena, 2006).

```
Data   : Number of iterations i_max
Result      : Solution x* ∈ X
f* ← ∞
For i=1,...,i_max do
  x ← GreedyRandomizedConstruction()
  f(x) ← LocalSearch(x)
  If f(x) < f* Then
    f* ← f(x)
    x* ← x
  End If
End For
```

Figure 4 – A basic GRASP for minimization problem (Resende and Ribeiro, 2005).

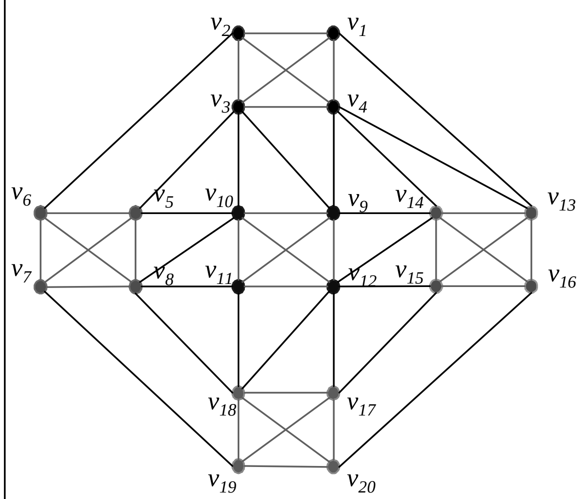
```
Procedure GreedyRandomizedConstruction()

Sol ← {}
V_Greedy ← V
While V_Greedy ≠ {} Do
 CreateRCL(RCL, V_Greedy, Sol)
 v ← ElementRandomizedSelection(RCL)
 Sol ← Sol ∪ v
 ReduceActiveVertices(V_Greedy, v)
End While

Return Sol
```

Figure 5 – A constructive heuristic for the PFCLP.

Initial Conflic Graph – First Iteration



First Step – Construction Phase

M = 10 in Equation (5)

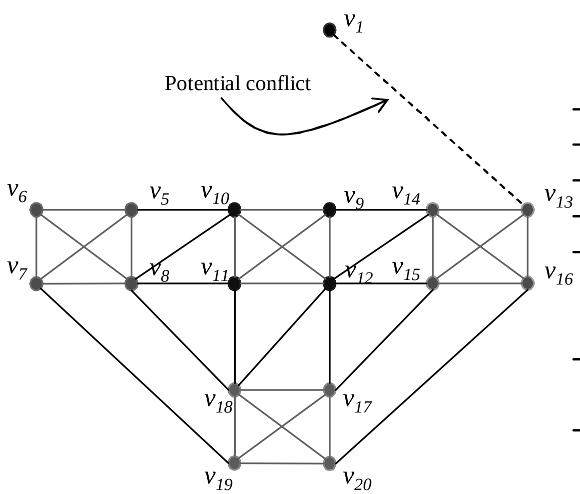
Vertices	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**
Cartographic preference	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Degree	4	4	6	6	5	4	4	6	6	6	5	7	5	6	5	4	5	6	4	4
Potencial conflicts	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Weight	**5**	**5**	**7**	**7**	**6**	**5**	**5**	**7**	**7**	**7**	**6**	**8**	**6**	**7**	**6**	**5**	**6**	**7**	**5**	**5**

	RCL																			
Sorted vertices	**1**	**2**	**6**	**7**	**16**	**19**	**20**	**5**	**11**	**13**	**15**	**17**	**3**	**4**	**8**	**9**	**10**	**14**	**18**	**12**

Selected vertex: 1 *Sol* = {1} Removing the candidate positions: {1, 2, 3, 4}

(a)

Reduced Conflic Graph – Second Iteration



Second Step – Construction Phase

M = 10 in Equation (5)

Vertices	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**
Cartographic preference	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Degree	4	3	4	6	4	5	5	7	3	5	5	4	5	6	4	4
Potencial conflicts	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
Weight	**5**	**4**	**5**	**7**	**5**	**6**	**6**	**8**	**14**	**6**	**6**	**5**	**6**	**7**	**5**	**5**

	RCL															
Sorted vertices	**6**	**5**	**7**	**9**	**16**	**19**	**20**	**10**	**11**	**14**	**15**	**17**	**8**	**18**	**12**	**13**

Select vertex: 5 *Sol* = {1, 5} Removing the candidate positions: {5, 6, 7, 8}

(b)

Figure 6 – An example of the constructive heuristic for the PFCLP.

```
Procedure LocalSearch(x)

// Let:
// - OF(Sol) be the objective function described in Equation (1)
// - NumberConflictFreeLabel(Sol)be a function that count the number of
//     conflict free labels presenting in feasible solution Sol

Sol ← x
FoundNewSolution ← true
fCurrent ← OF(Sol)
FreeLabels ← NumberConflicFreeLabel(Sol)
While FoundNewSolution Do
 FoundNewSolution ← false
 For i=1,…, N Do
   CurrentCandidatePosition = Sol[i]
   For ∀j∈P_i Do
    If j==Sol[i] Then Continue
      Sol[i] ← j
      If (OF(Sol)<fCurrent) Or
     (OF(Sol)==fCurrent And NumberConflicFreeLabel(Sol)> FreeLabels) Then
        fCurrent ← OF(Sol)
        FoundNewSolution ← true
        BestNeighbor ← j
        ChangedPoint ← i
        FreeLabels ← NumberConflicFreeLabel(Sol)
      End If
    End For
    Sol[i] ← CurrentCandidatePosition
  End For
  If FoundNewSolution Then
    Sol[ChangedPoint] ← BestNeighbor
  End If
 End While

x ← Sol
Return OF(Sol)
```

Figure 7 – Local search heuristic.

Table 1. Average results obtained with GRASP for *RCL_Length = 2*.

i_{max} = *100*								
Number of points	General average over 10 runs				Best average among 10 runs			
	Number of edges	Number of conflict labels	Proportion of conflict free labels (%)	Time (s)	Number of edges	Number of conflict labels	Proportion of conflict free labels (%)	Time (s)
25	3.00	5.13	79.50	0.023	3.00	5.13	79.50	0.023
100	0.00	0.00	100	0.003	0.00	0.00	100	0.003
250	0.00	0.00	100	0.029	0.00	0.00	100	0.028
500	0.96	1.80	99.64	8.203	0.96	1.80	99.64	8.199
750	9.84	18.88	97.48	40.410	9.84	18.88	97.48	40.386
1000	43.12	80.80	91.92	112.522	43.12	80.80	91.92	112.512

Table 2. Average results obtained with GRASP for *RCL_Length = 5*.

$i_{max} = 100$								
Number of points	General average over 10 runs				Best average among 10 runs			
	Number of edges	Number of conflict labels	Proportion of conflict free labels (%)	Time (s)	Number of edges	Number of conflict labels	Proportion of conflict free labels (%)	Time (s)
25	2.75	4.66	81.35	0.023	2.75	4.62	81.50	0.022
100	0.00	0.00	100.00	0.004	0.00	0.00	100.00	0.002
250	0.00	0.00	100.00	0.038	0.00	0.00	100.00	0.034
500	0.85	1.66	99.67	8.119	0.840	1.64	99.67	7.757
750	9.39	17.86	97.62	40.364	9.24	17.84	97.62	40.370
1000	42.98	81.43	91.86	116.856	42.40	81.08	91.89	112.49

Table 3. Average results obtained with GRASP for *RCL_Length = 6.*

$i_{max} = 100$								
Number of points	General average over 10 runs				Best average among 10 runs			
	Number of edges	Number of conflict labels	Proportion of conflict free labels (%)	Time (s)	Number of edges	Number of conflict labels	Proportion of conflict free labels (%)	Time (s)
25	2.79	4.80	80.80	0.026	2.75	4.62	81.50	0.025
100	0.00	0.00	100.00	0.004	0.00	0.00	100.00	0.002
250	0.00	0.00	100.00	0.049	0.00	0.00	100.00	0.036
500	0.848	1.66	99.67	9.055	0.84	1.64	99.67	8.268
750	9.36	17.84	97.62	42.25	9.20	17.68	97.64	40.363
1000	42.84	80.99	91.90	112.46	42.28	79.56	92.04	112.457

Table 4. Average results obtained with GRASP for *RCL_Length = 10.*

$i_{max} = 100$								
Number of points	General average over 10 runs				Best average among 10 runs			
	Number of edges	Number of conflict labels	Proportion of conflict free labels (%)	Time (s)	Number of edges	Number of conflict labels	Proportion of conflict free labels (%)	Time (s)
25	2.80	4.70	81.20	0.026	2.75	4.62	81.50	0.0215
100	0.00	0.00	100.00	0.003	0.00	0.00	100.00	0.002
250	0.00	0.00	100.00	0.052	0.00	0.00	100.00	0.034
500	0.90	1.72	99.66	8.831	0.88	1.60	99.68	8.149
750	9.87	18.92	97.48	40.397	9.72	18.68	97.51	40.401
1000	43.44	81.78	91.82	116.117	42.76	81.08	91.89	116.556

Table 5. Average results obtained with GRASP for *RCL_Length = 2* using the reduction proposed by Wagner et al (2001).

$i_{max} = 100$								
Number of points	General average over 10 runs				Best average among 10 runs			
	Number of edges	Number of conflict labels	Proportion of conflict free labels (%)	Time (s)	Number of edges	Number of conflict labels	Proportion of conflict free labels (%)	Time (s)
25	3.00	5.63	77.50	0.016	3.00	5.63	77.50	0.011
100	0.00	0.00	100.00	0.000	0.00	0.00	100.00	0.000
250	0.00	0.00	100.00	0.003	0.00	0.00	100.00	0.001
500	0.92	1.76	99.65	0.337	0.92	1.76	99.65	0.335
750	9.96	19.04	97.46	5.210	9.96	19.04	97.46	5.207
1000	43.12	81.28	91.87	26.823	43.12	81.28	91.87	26.673

Table 6. Average results obtained with GRASP for *RCL_Length = 5* using the reduction proposed by Wagner et al (2001).

$i_{max} = 100$								
Number of points	General average over 10 runs				Best average among 10 runs			
	Number of edges	Number of conflict labels	Proportion of conflict free labels (%)	Time (s)	Number of edges	Number of conflict labels	Proportion of conflict free labels (%)	Time (s)
25	2.75	4.68	81.25	0.018	2.75	4.625	81.50	0.015
100	0.00	0.00	100.00	0.000	0.00	0.00	100.00	0.000
250	0.00	0.00	100.00	0.002	0.00	0.00	100.00	0.000
500	0.85	1.65	99.67	0.362	0.84	1.64	99.67	0.346
750	9.13	17.28	97.70	5.214	9.04	16.96	97.74	5.216
1000	42.09	79.82	92.02	26.82	41.92	79.28	92.07	26.718

Table 7. Average results obtained with GRASP for *RCL_Length = 6* using the reduction proposed by Wagner et al (2001).

$i_{max} = 100$								
Number of points	General average over 10 runs				Best average among 10 runs			
	Number of edges	Number of conflict labels	Proportion of conflict free labels (%)	Time (s)	Number of edges	Number of conflict labels	Proportion of conflict free labels (%)	Time (s)
25	2.78	4.625	81.50	0.016	2.75	4.625	81.50	0.014
100	0.00	0.00	100.00	0.000	0.00	0.00	100.00	0.000
250	0.00	0.00	100.00	0.000	0.00	0.00	100.00	0.000
500	0.84	1.656	99.67	0.371	0.84	1.64	99.67	0.345
750	9.14	17.35	97.69	5.220	9.00	17.12	97.72	5.216
1000	41.88	79.388	92.06	26.788	41.60	77.96	92.20	26.688

Table 8. Average results obtained with GRASP for *RCL_Length = 10* using the reduction proposed by Wagner et al (2001).

$i_{max} = 100$								
Number of points	General average over 10 runs				Best average among 10 runs			
	Number of edges	Number of conflict labels	Proportion of conflict free labels (%)	Time (s)	Number of edges	Number of conflict labels	Proportion of conflict free labels (%)	Time (s)
25	2.75	4.79	80.85	0.017	2.75	4.75	81.00	0.016
100	0.00	0.00	100.00	0.000	0.00	0.00	100.00	0.000
250	0.00	0.00	100.00	0.000	0.00	0.00	100.00	0.000
500	0.86	1.68	99.66	0.376	0.84	1.64	99.67	0.347
750	9.42	18.03	97.60	5.230	9.40	17.88	97.62	5.217
1000	42.58	80.42	91.96	26.796	42.20	79.68	92.03	26.691

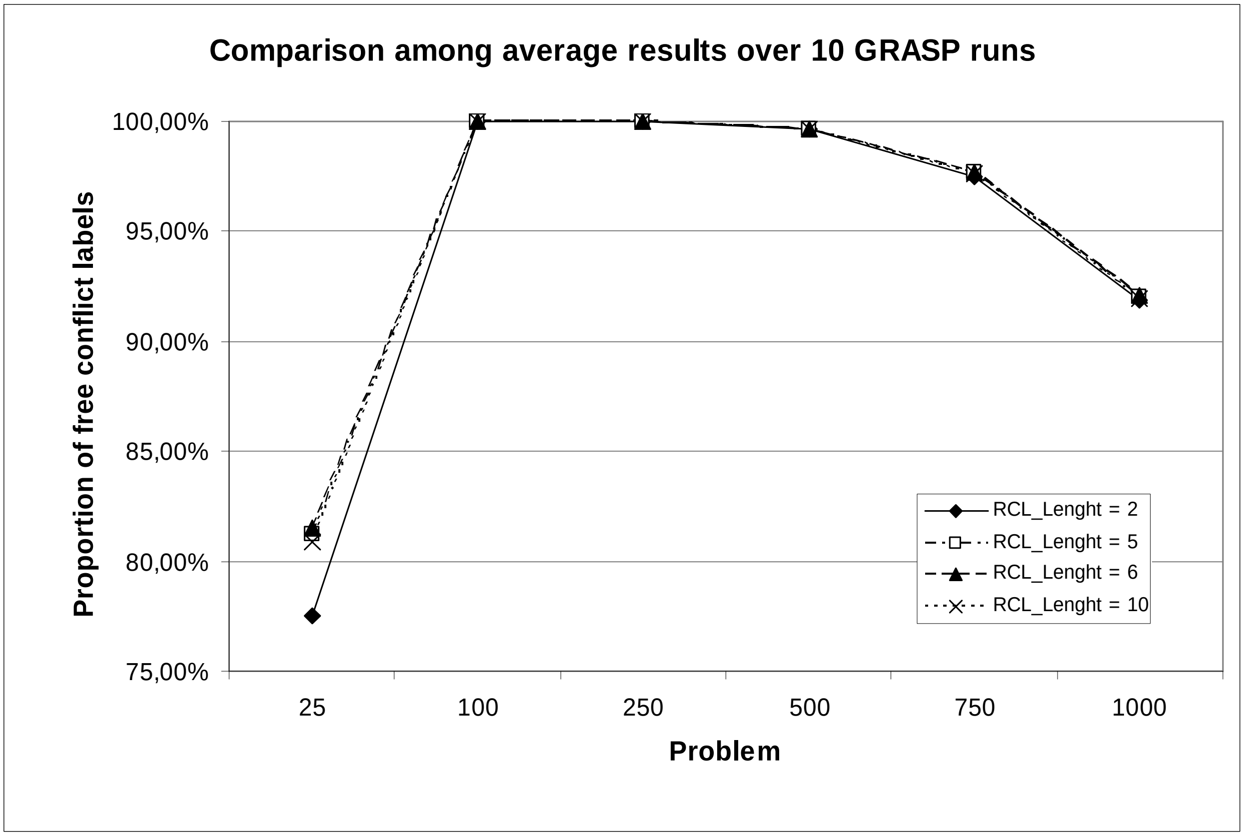


Figure 8 – Comparison among proportion of conflict free label considering the average results over 10 GRASP runs and the reduction proposed by Wagner et al (2001).

Table 9. Comparison with the literature

Algorithm	Proportion of conflict free labels				
	Problems				
	100	250	500	750	1000
$GRASP_{Best, RCL Length = 6}$	100.00	100.00	99.67	97.72	92.20
$GRASP_{Average, RCL Length = 6}$	100.00	100.00	99.67	97.69	92.06
$GRASP_{Best, RCL Length = 5}$	100.00	100.00	99.67	97.70	92.02
$GRASP_{Average, RCL Length = 5}$	100.00	100.00	99.67	97.74	92.07
LagClus (Ribeiro and Lorena, 2006)	100.00	100.00	99.67	97.65	91.42
CGA_{Best} (Yamamoto and Lorena, 2005)	100.00	100.00	99.60	97.10	90.70
FALP (Yamamoto et al., 2005)	100.00	100.00	99.50	96.70	90.12
$CGA_{Average}$ (Yamamoto and Lorena, 2005)	100.00	100.00	99.60	96.80	90.40
Tabu Search (Yamamoto et al, 2002)	100.00	100.00	99.30	96.80	90.00
GA with masking (Verner et al, 1997)	100.00	99.98	98.79	95.99	88.96
GA (Verner et al, 1997)	100.00	98.40	92.59	82.38	65.70
Simulated Annealing (Christensen et al, 1995)	100.00	99.90	98.30	92.30	82.09
Zoraster (Zoraster, 1990)	100.00	99.79	96.21	79.78	53.06
Hirsh (Hirsh, 1982)	100.00	99.58	95.70	82.04	60.24
3-opt Gradient Descent (Christensen et al, 1995)	100.00	99.76	97.34	89.44	77.83
2-opt Gradient Descent (Christensen et al, 1995)	100.00	99.36	95.62	85.60	73.37
Gradient Descent (Christensen et al, 1995)	98.64	95.47	86.46	72.40	58.29
Greedy Algorithm (Christensen et al, 1995)	95.12	88.82	75.15	58.57	43.41