An Hybrid Metaheuristic Approach to Solve the Euclidean Steiner Tree Problem in R^N

Marcelo Lisboa Rocha
Curso de Ciência da Computação - Universidade Federal do Tocantins
Av. NS 15, S/N – ALCNO 14 - Bloco 02 – Sala 21, CEP: 77020-210, Palmas - TO
marcelolisboarocha@gmail.com

ABSTRACT

Given a fixed set of points in a 3D space with Euclidean metric, the Tridimensional Euclidean Steiner Tree Problem consists of finding a minimum length tree that spans all these points using, if necessary, extra points (Steiner points). The finding of such solution is a NP-hard problem. This paper presents a hybrid metaheuristic based on GRASP and path relinking to the problem considered. Finally, computational experiments compares the performance of the proposed heuristic with previous works in the literature.

KEYWORDS. Euclidean Steiner Tree, GRASP, Path Relinking.

Main Area: Metaheuristics.

1. Introduction

The Steiner tree problem has been studied for a long time (Courant e Robbins, 1941), on its various aspects, such as: Steiner tree in graphs, rectilinear Steiner tree, Euclidean Steiner tree in plane (R^2) and more recently Euclidean Steiner tree in R^N among others.

This great interest in the Steiner tree problem is due to the fact that it has several practical and theoretical problems can be modeled as a Steiner tree problem. At present, a greater emphasis has been given to methodologies that efficiently search for solutions to the Euclidean Steiner Tree Problem (ESTP) in R^N, more specifically when $N = 3$ (tridimensional case) for applications, for example, in biochemistry (Smith and Toppur, 1996), phylogenetic inference (Montenegro et al., 2003) and design of networks in mines etc) (Alford et al., 2006). However, actually very few heuristics that can produce reasonable solutions are known and there is only two exact method applied just to tiny size instances (Smith, 1992) and (Fampa and Anstreicher, 2008).

This necessity for more efficient methods to ESTP in R^N (N dimensions considered) is the main motivations of this work, namely to provide a hybrid metaheuristic based on GRASP and path relinking to solve this problem. With the application of proposed hybrid metaheuristic, high quality near-optimal solutions are found, while seeking to maintain a satisfactory computational time (good performance).

This work, besides hybrid metaheuristic, also presented the results obtained from computational experiments. Following, are presented comparisons with the best techniques of literature.

This paper is organized as follows. Next section introduces ESTP formally and mentions the principal exact method in the literature known as Smith algorithm. Section 3 presents hybrid GRASP metaheuristic proposed, with its constructive and local search phases, well as path relinking method. Section 4 presents the experiments carried out as well as the computational results obtained, together with comparisons with other existing heuristics in the literature. Conclusions and future work are presented in Section 5.

2. Definition of Euclidean Steiner Tree Problem

The Euclidean Steiner Tree Problem (ESTP) is as follows: Given *P* points (also called obligatory) in R^N with Euclidean metric, find a minimum spanning tree (MST) that connect the given points, using, if necessary, extra points known as Steiner points, (see Figure 1). The remainder of this section presents the main characteristics of ESTP.

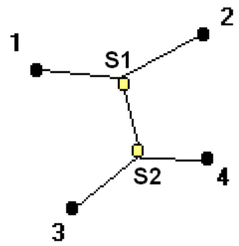


Figure 1: Steiner Tree with four obligatory points (*P*=4) and two Steiner points (*S1* and *S2*).

Suppose given *P* points $x^i \in R^N$, i = 1, 2, 3, ..., *P* in *N* dimensional space. Then, a solution of ESTP, called Steiner Minimum Tree (SMT) must present the following properties (Smith, 1992):

- The maximum number of Steiner points (*K*) is *P*-2;
- a Steiner point must have valence (or degree) equal to 3;
- edges emanating from a Steiner point lie in the same plane and have mutual angles of 120°.

If a tree (minimum or not) satisfies such properties, then we call it a Steiner tree. We call a Steiner topology the graph that represents a Steiner tree. The total number of different topologies with *K* Steiner points is $C_{P,K+2} = \frac{(P+K-2)!}{(k!2^K)}$. When *K*=*P*-2, we have a *Full Steiner Tree* (FST) and the number of different topologies is $f(P) = \frac{(2P-4)!}{2^{P-2}(P-2)!}$. Considering, for example, *P*=10, the total number of full topologies is *f*(10) = 2.027.025. This is the number of full topologies to be minimized by a brute force method. In Garey at al. (1977) and Garey and Johnson (1979), the ESTP is shown to be NP-Hard, fundamentally due to this combinatorial explosion, and is not possible to find an exact polynomial time algorithm for this problem.

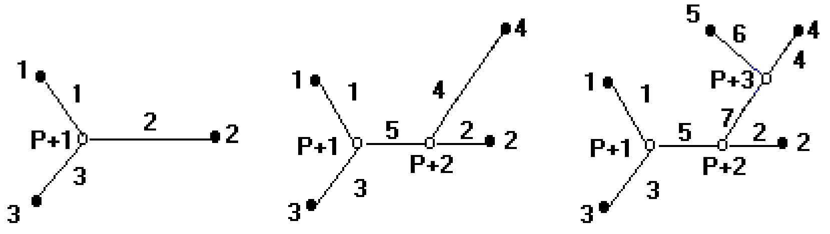
In the next subsection, more details will be given about full topologies to represent Euclidean Steiner trees and their importance in this work.

2.1. Describing Full Topologies

There exists an 1-1 (one-to-one) correspondence between full Steiner topology with $P \geq 3$ given points and (P – 3)-vectors, here denoted as *a* vectors. In each vector *a*, its i^{th} component (position) a_i corresponds to an integer value in the range $1 \leq a_i \leq 2i + 1$. One Steiner tree can be generated constructively given a (P – 3)-vector *a* as follows. Starting with an initial null vector ( ), corresponding to a full Steiner topology with three given points, 1, 2 and 3, connected through respective edges 1, 2 and 3 and one Steiner point numbered as *P*+1. This is clear because it is necessary at least three points to have a Steiner tree. After this step, all entries of the topology vector are considered, one at a time, where the i^{th} entry of topology vector is related to the insertion of *P*+i+1 Steiner point over the edge a_i and its connection with the given point i+3. This involves the addition of two new edges: the edge 2i+2, connecting the new Steiner point (*P*+i+1) to the new given point (i+3), and the 2i+3 edge, connecting this Steiner point (*P*+i+1) to the extreme (vertex) of a_i edge that have the larger number.

Figure 2 presents an example of construction of a full Steiner tree, using the topology vector *a* = (2, 4) and considering *P* = 5 and *N* = 3.

On the construction of the full Steiner tree corresponding to the topology vector *a* = (2, 4), the initial null vector ( ) is considered first, using the first three given points 1, 2 and 3 and the Steiner point, numbered as *P*+1. Next, considering the first element (i=1) of the topology vector, whose value is 2 (a_1=2), the Steiner point *P*+2 must be inserted in the edge 2 to connect the given point 4, generating two new edges numbered 4 and 5, as can be seen in Figure 2b. The second element (i=2) of the topology vector, whose value is 4 (a_2=4), indicates the edge where the Steiner point *P*+3 must be inserted to connect the given point 5, generating two new edges numbered 6 and 7, as can be seen in Figure 2c.



(a) (b) (c)

Figure 2: The initial null vector ( ) corresponds to the topology presented in (a); the topology with connection of given point 4 on edge 2 corresponding to the vector (2) is presented in (b), and the topology with the given point 5 inserted on edge 4 is presented in (c).

All trees of non-full topologies could be considered to be full Steiner topologies, with one or more Steiner point coinciding with given points. This interpretation, together with the use of adequate methods of minimization, makes it sufficient to focus on full topologies when searching for solution to TESTP. One efficient minimization method is proposed by Smith (1992) and will be approached in Section 2.2. Another existing exact method is the one proposed by Fampa and Anstreicher (2008), named Smith+, which is a improved version of Smith algorithm (1992). However, Fampa and Anstreicher (2008) method was not considered in this work , because both source-code as used test instances are not available and given the same has been applied only to small instances ($P \leq 18$).

2.2. Smith Algorithm

Smith (1992) proposes an enumerative method to find an exact solution to the ESTP in dimension $n \geq 3$, consisting basically of enumeration of all possible solutions to the problem and subsequent minimization, selecting the best solution among these. Enumeration of solutions is done through the use of topology vectors, each one representing a possible full topology, that are applied in a context of a *branch-and-bound* algorithm.

An important consideration about Smith algorithm is that it only searches full topologies (*FST*). The reason for this simplification is that the optimization method used by Smith allows considering other topologies as degenerations of complete topologies, where some Steiner points coincide with the given points. Despite this, the number of *FST* is still exponential in *n*.

Smith affirms that his algorithm is only capable of solving problems with at most twelve points, but in any dimension. This fact limits the practical application of the Smith algorithm. Despite this, Smith's (1992) algorithm is important since it opens up the possibility of use of its minimization step to find the optimal position of Steiner points for a given topology vector.

3. Hybrid Metaheuristic Method to Solve ESTP

This paper proposes a hybrid metaheuristic that mix together GRASP and path relinking to solve ESET. GRASP is a semi-greedy metaheuristic proposed by Resende and Feo (1995). This metaheuristic has two phases, first consist of a semi-greedy constructive phase and second is a local search. Here, an intensification strategy known as Path Relinking (PR) proposed by Glover (1996) was incorporated (hibridized) to GRASP, being justified by the success obtained by improving solutions found by GRASP (Laguna e Marti, 1999; Festa e Resende, 2008). This way, the following subsection describes constructive, local search and path relinking phases of the proposed hybrid metaheuristic, here called GRASP-PR.

3.1. Constructive Phase

In constructive phase, at each step, a given pont (fixed) is considered and a new Steiner point is inserted to the tree and its position, as the other Steiner point already considered, are optimize according to Smith criteria (1992), as cited in 2.2 section. Constructive steps are the following.

Step 1: Initial Step

- Generate all topologies of size 1 and calculate their respective lengths.
- Insert them in the Restricted Candidate List (RCL).
- Choose a topology in accordance with the RCL criteria.

Step 2: Repeat until all descendent topology vectors have size *P* - 3

- Generate all descendent topologies from one chosen in the previous step and calculate their respective lengths.
- Insert all generated topologies in RCL.
- According with RCL criteria choose one topology.

The constructive algorithm as cited is shown in Figure 3.

Phase Constructive
Initiate S = Ø ; //topologic vector
Initialize i = 1;
Define T = all topologies of size i;
While (i < P - 3) **Do**
  Calculate $V^- = \min\{V(t) : t \in T\}$; //V(t) is the cost of the tree of topology t
  Calculate $V^+ = \max\{V(t) : t \in T\}$; //V(t) is the cost of the tree of topology t
  Define RCL = $\{ t \in T : V(t) \leq V^- + \alpha \times (V^+ - V^-)\}$;
  Select $t \in$ RCL at random;
  i = i + 1;
  Define T = all topologies of size i descendents of t;
End While;
S = $\{t \in T : \min(V(t) : t \in T)\}$; //select the minimum size topology in T
Return S;

Figure 3: Constructive algorithm to ESET.

From described algorithm in Figure 3, lines 1 to 3 corresponds to Step 1 and lines from 4 to 10, corresponds to Step 2. An example of constructive phase with *P* = 6 and α = 1 is as follows..

Example to *P* = 6 and α = 1

- Step 1 generated the topologies (vectors) A(1), A(2) and A(3) and suppose that the chosen was A(2).
- In the first execution of step 2, all descendent topologies of the one chosen in the previous step (A(2)) are generated. They are: A(2,1), A(2,2), A(2,3), A(2,4) and A(2,5). Suppose that, in this step, the chosen is A(2,3).
- In the second execution of step 2, all descendent topologies of the chosen in the previous step (A(2,3)) are generated These topologies are: A(2,3,1), A(2,3,2), A(2,3,3), A(2,3,4), A(2,3,5), A(2,3,6) and A(2,3,7). Suppose that the chosen in this step is A(2,3,4). The last full topology chosen in step 2 (in this case A(2,3,4)), will be the initial solution for the local search phase.

3.2. Local Search Phase

Local search phase starts with a solution generated on constructive phase. Local search algorithm is presented in Figure 4.

```
Phase Local_Search (S) // S is the initial topologic vector
stop = value; //number of iterations with no improvement
cont = 0; initial topologic vector S; ρ = val(S);
Repeat
   For z=1 to numneighbor (can be P - 3) Do
      S' = S;
      i = rand(P-3); //sort position from 1 to P-3
      S'[i] = rand(2*i+1) //randomly choose a new value 1 ≤ S'[i] ≤ 2*i+1 different
               //from all others S'[i] (S'[i] ∩ S' = ∅)
      ρ' = val(S');
      cont = cont+1;
       If (ρ' < ρ) Then
         S = S';   ρ = ρ';   cont = 0;
       End If
   End For
   cont = cont + 1;
Until (cont >= stopBL);
Return S;
```

Figure 4: Local Search algorithm to the ESTP.

The local search phase uses the initial solution from the constructive phase and explores the neighborhood around this solution being this a topology descriptor (*P* – 3) – vector A, as cited in the constructive phase. The proposed method uses a simple neighborhood, where, given a topology vector A, its neighbors will be the topology vectors A' obtained by the change of only one of the *P* – 3 entries of A. For the proposed local search, the following consideration must to be taken:

- if an improvement is found, the current solution is updated and again a neighborhood around the new solution is searched.

- the process is repeated for a fixed number of times (stop variable), specified by the user.
- The number of neighbors (numneighbor) to be searched for a given topology is defined (empirically) as:
    - If $P \leq 100$ then numneighbor=P*0.5; else numneighbor=P*0.2;
    - This estimate is done in order that it is not necessary to search the whole neighborhood of a given topology, which would make the method too slow.

3.3. Path Relinking to ESTP

This section presents path relinking implementation to ESET. The algorithm starts from a T_0 solution and step-by-step transforms it in another solution T_d, where each solution is a topologic vector. In this path is possible to find a better solution than T_0 and T_d. Solution T_0, from where the method starts, is called base solution and T_d, the solution where is pretended to arrive, is called guide solution.

If T_0 and T_d are two solutions (topologic vectors) with *d* edges to insert different Steiner points, a movement from T_0 to T_d is a substitution of a Steiner point insertion edge in T_0 by a Steiner point insertion edge in T_d. In other words, taken a base solution T_0 and a guide solution T_2, there are two different edges to insert Steiner points (d = 2) between this two solutions. This way, there are two possibilities to intermediary solution T_1.

Path relinking method works in the following way: starting from T_0, a movement to T_d is generated and we have a intermediary solution T_1. Next, one more movement is generated, from T_1 to T_d, and so on until we have a solution T_{d-1}, after d-1 movements made.

Figure 5 presents how this method works. As T_0 is the topologic vector formed by insertion edges of Steiner points {2, 4} and the guide solution T_d is the topologic vector formed by insertion edges of Steiner points {1, 5}, we have two differences (d = 2) between this two solutions.

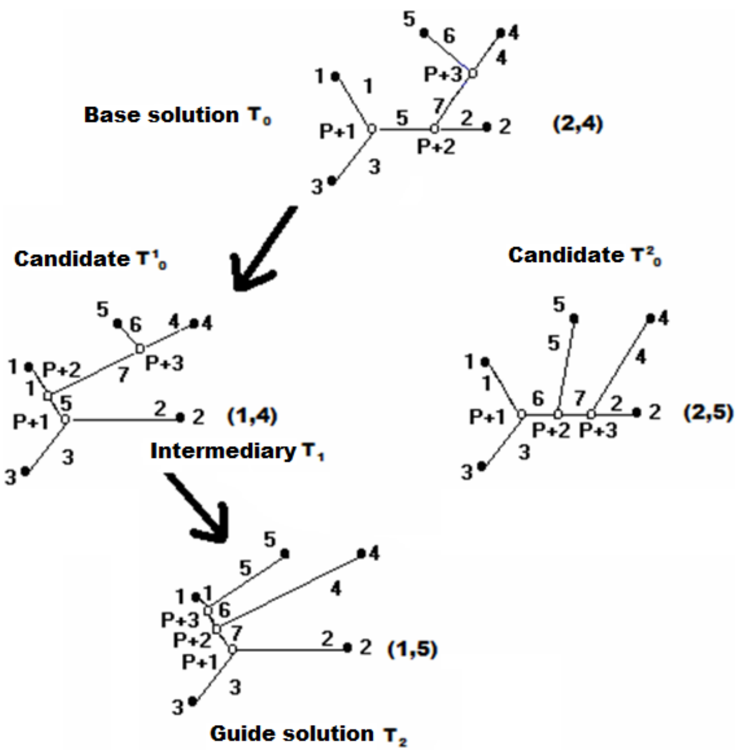


Figure 5: Example of how path relinking works to ESET.

This way, there are two candidates from T_0 to generate T_1 (this are T^1_0={1, 4} e T^2_0={2, 5}). Suppose the best option is change the insertion edge of Steiner points from 2 to 1. Thus, is generated the intermediate solution T_1 {1, 4}. The next movement, there is only one possibility, change the insertion edge of Steiner point 4 by 5. With this change, the intermediate solution T_2 is generated. As it is the same than T_d, the method stops.

3.4. Hybridizing GRASP with Path Relinking

Among several alternatives to make GRASP performance better, one that is having very big success is hybridizing it with path relinking method, that have as goal improve the quality of solutions obtained by GRASP (Resende e Ribeiro, 2005). Accordingly to Aiex et al. (Aiex and Resende, 2005) hybridizing GRASP with path relinking had taken to significant improvements on the quality of solutions obtained when compared to the both methods working alone.

Figure 6 presents the proposed algorithm (called GRASP-PR).

```
Algorithm GRASP-PR
f(S) ← 0; ConjElite ← ∅; MaxElite ← n ;
  For i = 1 to MaxIter Do
  .    Apply constructive phase to obtain a feasible
  .    feasible solution S ;
  .    Apply local search in S obtaining
  .    a new solution S';
  .    If |ContElite| < MaxElite Then
  .    .   Verify if S' ∈ ConjElite;
  .    .   If S' ∉ ConjElite Then
  .    .   .  Insert S' in ConjElite;
  .    .   End If
  .    End If
  .    Else
  .    .   Verify if S' is better than the worse
  .    .   of ConjElite;
  .    .   If S' is better Then
  .    .   .   Verify if S' ∈ ConjElite;
  .    .   .   If S' ∉ ConjElite Then
  .    .   .   .   Insert S' in ConjElite;
  .    .   .   End If
  .    .   End If
  .    End Else
  .  Apply path_relinking in S';
  .  If value of f(S*) > f(S') Then
  .  .  S* ← S';
  .  End If
  End For
  Return S*;
End Algorithm
```

Figure 6: Algoritmo GRASP com *path-relinking*.

In GRASP-PR, user defines the number of executions (*MaxIter*) and the size of elite set

(*MaxElite*). After this, is executed the constructive phase (line 3). After, is executed the local search phase (line 4). Using orientations provided in Resende e Ribeiro (2005), path relinking had the following features:

- Start the procedure considering as base solution one elite solution selected at random and the guide solution, the solution S' provided by local search.
- Size of elite set (*MaxElite*) is 5.
- Path relinking methos is used (line 20) always that the solution S' (provided by local search) is *p*% worse that the best solution of elite set (*ConjElite*). Here, *p*=*iter*, where *iter* is the number of iterations with no update in the elite set. Thus, the larger the number of iterations without updating the elite set (*iter*), greater the probability of performing path relinking.
- Apply path relinking to the solutions S' and solution *CE* of elite set that maximize *d* (the most different), being base solution, the best between both.

Given that path relinking method is computational expensive to be applied to each GRASP iteration, is important to consider ways to avoid unnecessary effort. The way used in this work was the addition of memory. In this case, all pairs of solutions used as base and guide are stored in a *hash* table. Thus, a new pair of solutions (base and guide) will only be submited to path relinking if this is not already in *hash* table. *Hash* technique used in this work was with *hash* function by multiplication method and collision treatment by chaining. Another artifice used was to use a filtering strategy. In this case, a solution is only used as a base solution in the case that is maximum at λ% worse than the incumbent solution (best solution found). In this work, considered λ=1%, based on results obtained by Martins *et al.* (2000).

4. Preliminary Results Obtained to ESET

This section will be presented preliminary results of hybrid metaheuristic GRASP-PR proposed done over three set of 15 instances each with with *P* equal 50, 100 and 250 respectively in R^3 and generated as in Montenegro *et al.* (2001).

Here, the computational results obtained by GRASP-PR are compared to the presented in Rocha *et al.*, (2007). In Rocha *et al.*, (2007) are performed computational tests with several heuristics and and exact method. The heuristic methods are: Soap Film (Chapeau-Blondeau, Janez e Ferrier, 1997), Genetic Algorithms (GA) (Montenegro e Maculan, 2000), Microcanonical Otimization (μO) (Montenegro *et al.*, 2001) and GRASP (Rocha *et al.*, 2007) called GRASP-S; more the exact method proposed in Smith (1992). Among all, GRASP-S stood out, obtaining best performance, well relative to the quality of solutions found as computational time. This way, is the literature's technique considered to comparison. To evaluate the quality of solutions found by the heuristic methods, was used a reference measure known as Steiner Ratio (ρ), defined as $\rho = L_{SMT}/L_{MST}$, where L_{SMT} is the length of Steiner Minimum Tree (exact or heuristic) and L_{MST} is the length of Minimum Spanning Tree, which can be obtained in polynomial time (Prim, 1957). A heuristic solution is better the lower its ρ value.

The hybrid metaheuristic GRASP-PR proposed in this work was implemented using C programming language and compiled with *gcc 3.3.2* using the -O3 compiler option. All tests were done on a machine with the following configuration: Pentium Core Duo de 1,86 Ghz (code name *yonah* and 2 Mb of L2 cache), 2 Gb of RAM and operational system Linux Ubuntu 11.10. GRASP-PR was performed at the same machine and with the same parameters of GRASP-S. Path relinking parameters as cited in section 3.4 have their values presented in Table 1.

Table 1: Parameter values to GRASP-PR.

Parameter	α	*MaxIter*	*stopBL*	*MaxElite*	*p*	*λ*
Value	[0.1; 0.3]	100	10	5	*MaxIter*	1%

Computational results of GRASP-PR are presented in Table 2. Analyzing presented result in Table 2, GRASP-PR had a little increase in computational time, but obtained a greater number of better solutions found (NBSF) to instances with *P* equal 100 and 250, obtaining 3 better solutions in 15 to the first case and 5 better solutions in 15 to the second case than GRASP-S. To *P*=50 GRASP-S and GRASP-PR obtained the same solutions. In this two set (*P*=100 and *P*=250), mean reduction of obtained Steiner tree were approximately of 0,98% and 0,63% respectively.

Considering execute GRASP-PR having as target solutions the ones provided by GRASP-S, Figures 7, 8 and 9 illustrates the number of instances that each algorithm has reached the target per unit time for each set of instances (*P* = 50, *P* = 100 and *P* = 250).

From figures 7, 8 and 9 shows the impact of using path relinking procedure on the efficiency of GRASP heuristic, because it is found that the number of targets encountered with GRASP-PR is always greater at any instant of time considered that afforded by the GRASP-S in all sets of instances. An interesting detail to note is that the GRASP-PR hybrid metaheuristic can always find 50% or more of the targets within 10 seconds of running on three sets of instances. For the set of specified targets, the average time of the GRASP-PR for all three instances was 40.66 seconds vs. 51.33 seconds GRASP-S.

Table 2: Performance of GRASP-S and of GRASP-PR to the three set higher dimensionality instances.

Method		*P*		
		50	100	250
GRASP-S	Mean ρ	0,940796	0,948964	0,965472
	Standard Deviation	0,021645	0,022035	0,029665
	Mean Time	21,374	41,503	120,581
	NBSF	**0**	0	0
GRASP-PR	Mean ρ	0,940796	0,939641	0,959331
	Standard Deviation	0,021645	0,021885	0,024466
	Mean Time	23,593	45,834	130,458
	NBSF	**0**	**3**	**5**

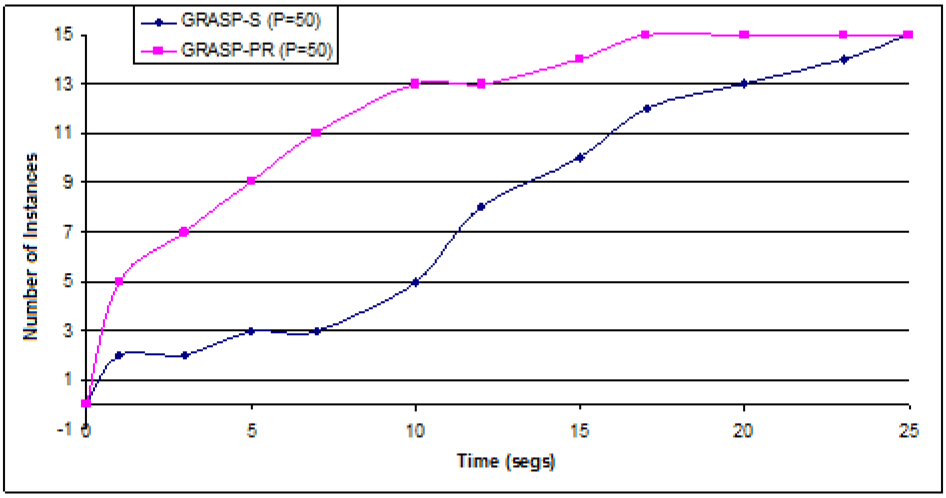


Figure 7: Number of targets found by GRASP-S and GRASP-PR to the set of instances *P*=50.

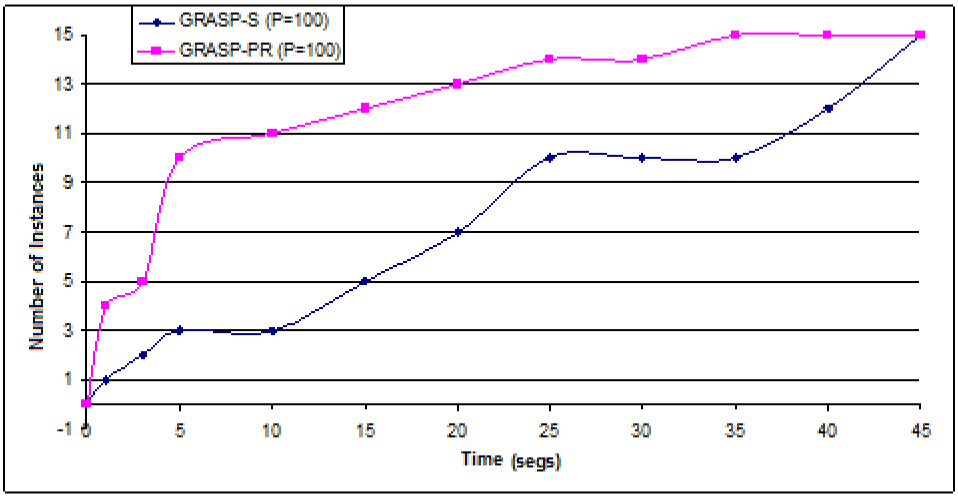


Figure 8: Number of targets found by GRASP-S and GRASP-PR to the set of instances *P* =100.

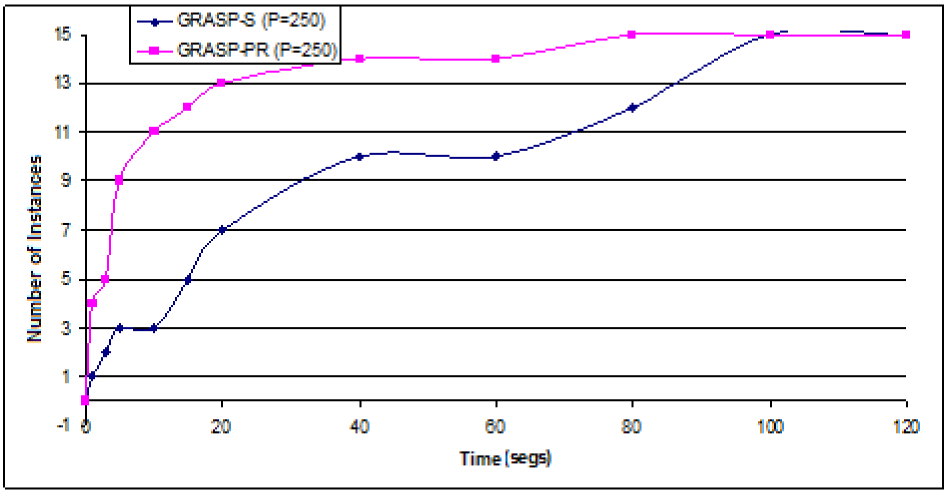


Figure 9: Number of targets found by GRASP-S and GRASP-PR to the set of instances *P* =250.

5. Conclusion and Future Works

This work presented an approach based on heuristic GRASP with reconnection paths to the Euclidean Steiner Tree Problem (ESTP) as a hybrid metaheuristic.

Computational experiments were also carried out and comparisons with the best technical literature for the size of the problems addressed, where tests have shown that the proposed GRASP-PR is very efficient for obtaining good solutions. The computational tests indicate that hybridization of GRASP heuristic with path relinking (GRASP-S more path relinking), here called GRASP-PR, provides a significant improvement in the performance. In this case, GRASP-PR obtained in various situations, better solutions than GRASP-S. Taking into account the achievement of targeted solutions, GRASP-PR also showed better performance with respect to execution time. This supports the concept that the hybridization process, when done well, it provides an improvement in the overall performance of the developed hybrid method, being superior to each party individually.

The computational results demonstrate the robustness and efficiency of the proposed hybrid metaheuristic, for finding the optimal solution for each problem set a greater number of times and to a lesser computational time than the competing techniques.

Experiments performed with GRASP-PR heuristic proposal for ESTP are very promising. However, further studies and new developments can be made in order to seek even greater efficiency for GRASP. Main suggestions have been:

- Perform computational tests with instances with higher number of data points (*P*).
- Implement this hybrid metaheuristic in parallel in order to increase the performance in terms of reduction in execution time of the method.
- make use of other techniques in the hybridization process and then evaluate the impact of use them on the performance of the proposed approach.

References

Aiex, R. M. and Resende, M. G. C. (2005): "Parallel strategies for GRASP with path-relinking". In: *Metaheuristics: Progress as Real Problem Solvers*, T. Ibaraki, K. Nonobe and M. Yagiura, (Eds.), Springer, pp. 301-331.

Alford, C., Brazil, M., Lee, D. H., 2006. Optimization in Underground Mining. In: *Handbook on Operations Research in Natural Resources*, Dordrecht/Boston/London: Kluwer Academic Publisher.

Chapeau-Blondeau, F., Janez, F. and Ferrier, J.-L. (1997): "A Dynamic Adaptive Relaxation Scheme Applied to the Euclidean Steiner Minimal Tree Problem". In: *SIAM Journal of Optimization* 7, 1037-1053.

Courant, R. and Robbins, H. (1941): *What is Mathematics?* Oxford University Press, New York.

Fampa, M. H. C. and Anstreicher, K. (2008) "An improved algorithm for computing Steiner minimal trees in Euclidean d-space", *Discrete Optimization* 5(2), pp. 530-540.

Festa, P. e Resende, M.G.C. (2008) "Hybrid GRASP heuristics. In "Global optimization: Theoretical foundations and applications", A. Abraham, A.-E. Hassanien, and P. Siarry (Eds.), *Studies in Computational Intelligence*, Springer-Verlag, 2008.

Garey, M. R., Graham, R. L. and Johnson, D. S. (1977): "The Complexity of Computing Steiner Minimal Tree". In: *SIAM Journal of Applied Mathematics* **32**, 835-859.

Garey, M. R. and Johnson, D. S. (1979): *Computers and Intractability: A Guide to the Theory of NP-completeness*. W. H. Freeman, San Francisco, USA.

Glover, F. (1996): "Tabu search and adaptive memory programming – Advances, applications and challenges". *Computing Tools for modeling, Optimization and Simulation: Interfaces in Computer Science and Operations Research*. Kluwer Academic Publishers, p. 1-75.

Laguna, M., Marti, R.. "Grasp and Path Relinking for 2-Layer Straight Line Crossing Minimization**"**. *INFORMS Journal on Computing*, v.11 n.1, p.44-52, 1999.

Martins, S. L., Resende, M. G. C., Ribeiro, C. C., and Pardalos, P. M. (2000): **"**A parallel GRASP for the Steiner tree problem in graphs using a hybrid local search strategy**"**. J. of Global Optimization, vol. 17, pp. 267-283.

Montenegro F., Maculan, N. (2000): "A Genetic Algorithm for the Euclidean Steiner Problem in R^n". Presented at *Proc X. Cong. Ibero-Latinoamericano Inv. Oper. A241*.

Montenegro, F., Maculan, N. and Torreão, J. R. A. (2001): "Microcanonical Optimization Applied to the Euclidean Steiner Problem in R^n". Presented at *The Fourth Metaheuristics International Conference (MIC2001)*, Porto, Portugal, July 2001.

Montenegro, F., Toreão, J. R. A., Maculan, N., 2003. Microcanonical Optimization Algorithm for the Euclidian Steiner Problem in R(n) with application to Phylogenetic Inference. *Physical Review E - Statistical Physics, Plasmas, Fluids and Related Interdisciplinary Topics*, v. 68, n. 5, p. 567.

Prim, R. C. (1957): "Shortest Connection Networks and Some Generalizations". In: *Bell Systems. Technical Journal* **36**, 1389-1401.

Resende, M. G. C. and Feo, T. A. (1995): "Greedy Randomized Adaptative Search Procedures". In: *Journal of Global Optimization* **6**, 109-133.

Resende, M. G. C. and Ribeiro, C. C. (2005): "GRASP with path-relinking: Recent advances and applications". In: Ibaraki, T., Nonobe, K. and Yagiura, M. (eds.): *Metaheuristics: Progress as Real Problem Solvers*. Springer, 29-63.

Rocha, M. L.; Bhaya, A.; Montenegro, F. M. T. and Maculan, N. (2007): "A GRASP Heuristic for the Tridimensional Euclidean Steiner Tree Problem". *Seventh Metaheuristic International Conference (MIC2007)*, Montreal, Canada, June 2007.

Smith, J. and Toppur, B. (1996): "Euclidean Steiner Minimal Trees, Minimum Energy Configurations, and the Embedding Problem of Weighted Graphs in E^3". In: *Discrete Applied Mathematics* 71, 187-215.

Smith, W. D. (1992): How to Find Steiner Minimal Trees in Euclidean d-space. Algorithmica, v.7, n. 2/3, pp.137-177.