Variable Neighborhood Descent with Iterated Local Search for Routing and Wavelength Assignment

Alexandre Martins, Christophe Duhamel, Philippe Mahey, Rodney R Saldanha, Mauricio C de Souza

Variable Neighborhood Descent with Iterated Local Search for Routing and Wavelength Assignment

Alexandre X. Martins* Christophe Duhamel† Philippe Mahey‡
Rodney R. Saldanha§ Mauricio C. de Souza¶‖

October 27, 2011

Abstract

In this work we treat the Routing and Wavelength Assignment (RWA) with focus on minimizing the number of wavelengths to route demand requests. Lightpaths are used to carry the traffic optically between origin-destination pairs. The RWA is subjected to wavelength continuity constraints, and a particular wavelength cannot be assigned to two different lightpaths sharing a common physical link. We develop a Variable Neighborhood Descent (VND) with Iterated Local Search (ILS) for the problem. In a VND phase we try to rearrange requests between subgraphs associated to subsets of a partition of the set of lightpath requests. In a feasible solution, lightpaths belonging to a subset can be routed with the same wavelength. Thus, the purpose is to eliminate one subset of the partition. When VND fails, we perform a ILS phase to disturb the requests distribution among the subsets of the partition. An iteration of the algorithm alternates between a VND phase and a ILS phase. We

*Programa de Pós-Graduação em Engenharia Elétrica, Universidade Federal de Minas Gerais, Av. Antônio Carlos, 6627, cep : 31270-901, Belo Horizonte, MG, Brasil, and Departamento de Ciências Exatas e Aplicadas, Universidade Federal de Ouro Preto, João Monlevade, MG, Brasil. e-mail : xmartins@decea.ufop.br

†Laboratoire LIMOS, CNRS-UMR6158, and ISIMA, Université Blaise Pascal, Campus des Cézeaux, BP 10125, 63173 Aubière CEDEX, France. e-mail : christophe.duhamel@isima.fr

‡Laboratoire LIMOS, CNRS-UMR6158, and ISIMA, Université Blaise Pascal, Campus des Cézeaux, BP 10125, 63173 Aubière CEDEX, France. e-mail : philippe.mahey@isima.fr

§Departamento de Engenharia Elétrica, Universidade Federal de Minas Gerais, Av. Antônio Carlos, 6627, cep : 31270-901, Belo Horizonte, MG, Brasil. e-mail : rodney@cpdee.ufmg.br

¶Departamento de Engenharia de Produção, Universidade Federal de Minas Gerais, Av. Antônio Carlos, 6627, cep : 31270-901, Belo Horizonte, MG, Brasil. e-mail : mauricio.souza@pq.cnpq.br

‖corresponding author

report computational experiments that show VND-ILS was able to improve results upon powerful methods proposed in the literature.

Keywords : Routing and assignment; Network design; Variable neighborhood descent; Local search

1 Introduction

Optical networks with wavelength division multiplexing (WDM) provide users with very large bandwidths. In all-optical networks a traffic demand is carried from source to destination through a lightpath, which is a sequence of fiber links carrying the traffic optically from end to end [5]. The wavelength continuity constraint implies that to a given lightpath a single wavelength must be assigned, i.e., a particular wavelength must be reserved to travel with the traffic on each link the lightpath traverses. Moreover, a particular wavelength cannot be assigned to two different lightpaths sharing a common physical link.

The Routing and Wavelength Assignment (RWA) problem deals with the routing and the assignment of wavelengths to lightpath requests between pairs of nodes. Given a set of lightpath requests, two variants of the RWA problem have been studied in the literature: `max`-RWA and `min`-RWA. In the former, the objective is to maximize the number lightpath requests that can be routed with a fixed number of wavelengths. In the latter, the objective is to minimize the number of wavelengths to route all the requests.

In this paper, we develop a Variable Neighborhood Descent (VND) with an Iterated Local Search (ILS) based perturbation for the `min`-RWA. Let $G = (V, E)$ be a digraph where V is the set of nodes and E is the set of bidirectional arcs. We denote by Γ the set of lightpath requests where each $r \in \Gamma$ is defined by an origin and destination pair $(s_r, d_r) \in V \times V$. Note that we can have two different requests with the same origin-destination pair and in this case if they are to be routed with the same wavelength, then they have to be routed through two arc disjoint paths in G. The problem is to find a minimal partition of Γ in W subsets such that the requests in each Γ_w, $w = 1, \ldots, W$, can be routed through arc disjoint paths in G.

The paper is structured as follows. In the next section we discuss related works in the literature. In Section 3 we describe the VND-ILS heuristic for the `min`-RWA. Then, in Section 4 we report numerical results on some hardest benchmark instances from the literature. We end with concluding remarks and extensions for future work.

2 Related works

Chlamtac et al. [4] have shown that the RWA problem is NP-Complete, and proposed, to our knowledge, the first greedy heuristics for the problem. Dutta and Rouskas [5] and Zang et al. [26] reviewed the literature on the RWA problem covering different approaches and variants developed in the 90's. For instance, Ramaswami and Sivarajan [23] derived upper bounds on the number of lightpath requests that can be routed, while in [24] they proposed to minimize the network congestion subjected to average delay constraints. Banerjee and Mukherjee [3] partitioned large RWA problems into several smaller subproblems to be solved independently and efficiently. Mukherjee et al. [19] proposed to alternate two phases: simulated annealing to construct a topology, and flow deviation to route the traffic. See also Kennington et al. [12] for the RWA problem in survivable WDM networks.

Mathematical formulations have been developed for the RWA problem, with particular emphasis on path-based ones. Ramaswami and Sivarajan [23] first modeled the RWA problem using a path-based formulation. Krishnaswamy and Sivarajan [13, 14] presented a mixed integer linear formulation which takes into account the maximum number of hops, among other logical and physical constraints, to minimize congestion. The authors developed heuristics based on rounding the solutions obtained by solving the respective LP-relaxation. Lee et al. [15] proposed a column generation algorithm to deal with the exponential number of variables associated to potential feasible paths. Jaumard et al. [9, 10], in comprehensive surveys, reviewed existing column generation formulations and proposed new ones. In particular, Jaumard et al. [11] developed improvements to make column generation methods efficient on the RWA problem. In their work, several instances were solved to optimality for the first time.

Some studies and tests have been done in order to design new or improved heuristics. The heuristics described below were developed for the `min`-RWA version of the problem. Manohar et el. [17] presented a greedy algorithm to do a partition of lightpath requests into subsets, each of which associated to a wavelength. Their algorithm explores techniques employed to solve the maximum edge disjoint paths problem. Noronha and Ribeiro [22] used a decomposition scheme in two distinct phases. In the first phase, a number of alternative routes are computed for each lightpath request. A conflict graph is built to exploit efficient heuristics for coloring problems. A route computed generates a node in the conflict graph, and an edge is set up between two nodes whose routes share a common link in the original graph. The set of nodes of the conflict graph is partitioned such that each subset contains the routes computed for a given lightpath request. Noronha and Ribeiro [22] proposed a tabu search for the partition coloring problem which is applied, in the second phase, over the conflict graph to assign wavelengths to routes.

Skorin-Kapov [25] adapted ideas from bin packing heuristics to the `min`-RWA. For such purpose, she considered lightpath requests as items, and copies of the original graph as bins. The equivalent for the weight of an item is the number of links to route a lightpath. To say that a bin has not enough capacity to accommodate two items is equivalent to not being able to route two lightpath requests on a copy of the original graph without sharing a link. According to the bin packing analogy Skorin-Kapov [25] proposed four variants of best fit and first fit heuristics. In the numerical results reported in [25] the best fit decreasing (BFD) was the most successful among them. Noronha et al. [20] improved performance of best fit and first fit heuristics. To do this, the authors worked on data structures and implementation strategies, such as double linked adjacency lists and dynamically updating of shortest paths. Noronha et al. [20] introduced a broader set of testbed instances, and maximum running times were reduced to one quarter of those with a standard implementation of BFD (which was confirmed as the heuristic to perform best).

More recently, Noronha et al. [21] embedded BFD into a biased random-key genetic algorithm. The chromosomes are vectors of real numbers, denoted keys, in the interval [0,1]. The keys are used to bias a decoding heuristic in generating a feasible solution. Actually, each key is associated with a lightpath. Lightpaths are sorted in non-increasing order of the sum of their lengths and keys, and then BFD is applied. Computational experiments were conducted on the most studied instances and as well on the new benchmark ones introduced in [20]. The genetic algorithm improved upon results from heuristics previously proposed in the literature. Noronha et al. [21] reported that it reached solutions better than or similar to those found by a multistart variant of BFD in 23% less time on average, and reduced on average to half gaps provided by the decomposition scheme proposed in [22].

3 VND-ILS heuristic for the min-RWA

A feasible solution is characterized by a partition of Γ in W subsets along with arcs disjoints paths to route requests belonging to each Γ_w, $w = 1, \ldots, W$, in G. Let us define by $F_r \subseteq E$ the arcs of the path used to route a request $r \in \Gamma$ in G. Each $\Gamma_w \subseteq \Gamma$ induces a subgraph $G_w = (V, E_w)$ of G where $E_w = E - \cup_{r \in \Gamma_w} F_r$. In other words, G_w contains the arcs not used to route requests belonging to Γ_w.

Given a feasible solution, in a VND phase we employ three kinds of moves trying to rearrange requests between the subgraphs associated to subsets of the partition in attempt to eliminate one of them. When VND fails, we perform a ILS-based perturbation phase to disturb the requests distribution among the subsets of the partition. Thus, an iteration of the algorithm alternates between a VND phase and a

ILS-based perturbation phase.

3.1 VND phase

VND is a search heuristic proposed by Mladenović and Hansen [18] within the framework of variable neighborhood search methods, see Hansen et al. [6, 7, 8]. The VND works with k_{max} neighborhood structures N_k, $k = 1, \ldots, k_{max}$, designed for a specific problem. It starts with a given feasible solution as incumbent and sets $k = 1$. If an improvement is obtained within neighborhood N_k, the method updates the new incumbent and sets $k = 1$. Otherwise, it increases the value of k and the next neighborhood is considered. The method stops when a local optimum for $N_{k_{max}}$ is found.

We propose a VND algorithm with $k_{max} = 3$. Let $\Gamma_{\bar{w}}$ be a subset contained in the partition characterizing a feasible solution. We consider two alternatives to chose $\Gamma_{\bar{w}}$. In the first alternative, denoted by VND$_r$, $\Gamma_{\bar{w}}$ is the subset of the partition with the least number of requests, i.e., $\bar{w} = \arg\min\{|\Gamma_w| : w = 1, \ldots, W\}$. The idea is that it might be easier to empty by reallocating requests a set with a few of them. In the second one, denoted by VND$_e$, $\Gamma_{\bar{w}}$ is the subset whose induced graph $G_{\bar{w}}$ has the greatest number of arcs, i.e., $\bar{w} = \arg\max\{|E_w| : w = 1, \ldots, W\}$. If a fewer number of arcs were used to route requests belonging to $\Gamma_{\bar{w}}$ then it might be possible to find paths to route them in the subgraphs induced by the other sets of the partition. In both alternatives, the requests belonging to $\Gamma_{\bar{w}}$ form a list $L_{\bar{w}}$. The VND traverses $L_{\bar{w}}$ and tries with each request r to perform moves within neighborhoods N_1, N_2, N_3 until it either succeeds to reallocate all requests belonging to $\Gamma_{\bar{w}}$ or fails to reallocate a request within N_3. In the former case, VND has emptied subset $\Gamma_{\bar{w}}$ of the partition, and consequently reduced the number of wavelengths to be used. In the latter case, the ILS-based perturbation phase is called. We remark that an improving move, if it happens, occurs only when the last remaining request of $\Gamma_{\bar{w}}$ is reallocated to another subset, while for $|\Gamma_{\bar{w}}| \geq 2$ performing a move does not reduce the objective function's value. When VND succeeds to empty $\Gamma_{\bar{w}}$, we update the remaining subsets of the partition with respect to the criteria to select the next subset to which VND is to be applied, according to the alternatives VND$_r$ or VND$_e$.

Suppose, traversing $L_{\bar{w}}$, the search is to consider a request r. A move in the first neighborhood N_1 tries to reallocate r to another subset of Γ. If the neighborhood N_1 is not empty with respect to a partition and a request, then there exists $\Gamma_{w'} \subset \Gamma$, $w' \neq \bar{w}$, whose induced graph $G_{w'}$ has a path between s_r and d_r (the request's origin and destination pair). In this case the search within N_1 stops when the first graph $G_{w'}$ to accommodate r is found, and r is transferred from $\Gamma_{\bar{w}}$ to $\Gamma_{w'}$. The search then continues to the next request in $L_{\bar{w}}$, or reduces a wavelength if r was the last of $L_{\bar{w}}$. The list $L_{\bar{w}}$ is sorted in non-increasing order of shortest path lengths in G

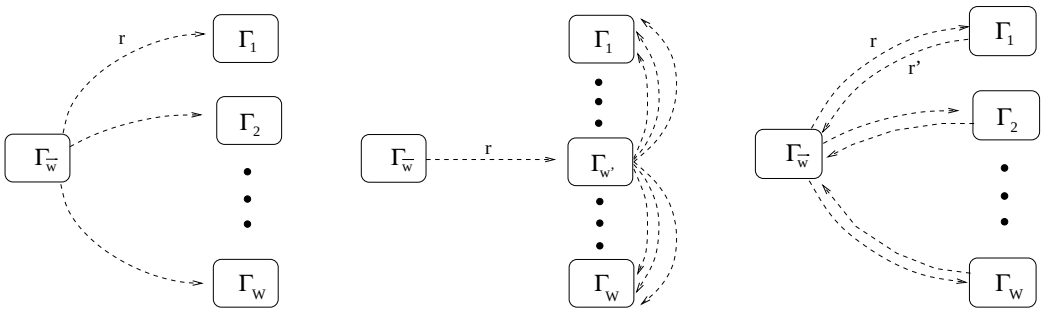
between origin and destination nodes. The motivation behind is to try to route in the subgraphs induced by the other sets first the requests using more arcs, and leave the requests using fewer arcs to try after since when a path is found to route a request in a subgraph its arcs are removed. If N_1 is empty, VND proceeds to neighborhood N_2 with the same request r under analysis.

The rationale in the second neighborhood is try first to make room for r in a graph $G_{w'}$ by transferring as many requests as possible from $\Gamma_{w'}$ to other subsets of Γ not equal to $\Gamma_{\bar{w}}$. This is done analogously to the search in neighborhood N_1, i.e., we traverse $L_{w'}$ and try, for each request, to find a subset $\Gamma_{\hat{w}}$, $\hat{w} \neq w'$ and $\hat{w} \neq \bar{w}$, such there is a path to route the request in $G_{\hat{w}}$. If we are able to unload at least one request from $\Gamma_{w'}$, then we can try to allocate r to it. Upon success, the search resumes from the next request in $L_{\bar{w}}$, if any, within neighborhood N_1. Otherwise, VND proceeds to neighborhood N_3 still with the same request r under analysis. Assume that the subsets of the partition were built in a given order, $w = 1, \ldots, W$. At each new search within N_2 we take only one subset of the partition, according to the order, to try to make room for r in it. That is, if we fail, during a search within N_2, to make room in a graph $G_{w'}$ for request r, we do not try to perform a move considering another graph but proceeds to neighborhood N_3. Then, the next time we proceed to N_2 the graph to be considered will be $G_{w'+1}$ (assuming $\bar{w} \neq w' + 1$).

A move in neighborhood N_3 seeks to swap request r with another request belonging to a subset of Γ different from $\Gamma_{\bar{w}}$. Given a request u, let $min_{spl}(u)$ be the number of arcs of the shortest path between s_u and d_u in G. The search considers requests with min_{spl} smaller than $min_{spl}(r)$, sorted in non-decreasing order of min_{spl}. The choice to consider only requests whose shortest paths in G have fewer arcs than r carries the idea that it might be easier to route such a request in other subgraphs of the partition when the search resumes from N_1. Thus, if a request $r' \in \Gamma_{w'}$, such that r can be routed in the subgraph induced in G by $\Gamma_{w'} - \{r'\}$, and r' in the one by $\Gamma_{\bar{w}} - \{r\}$, is found, then a swap between r and r' is performed. After a successful move in N_3 we have $r' \in \Gamma_{\bar{w}}$ and $r \in \Gamma_{w'}$. In this case, the search resumes by trying to transfer r' from $\Gamma_{\bar{w}}$ to another subset of the partition with a move in N_1. Note that a move in N_3 does not reduce $|\Gamma_{\bar{w}}|$, but since it swaps r with r', and $min_{spl}(r') < min_{spl}(r)$, it may allow a reallocation of r' with a move in N_1 in the sequel. If it is not possible within N_3 to swap r with any other request having a smaller min_{spl}, the search calls a ILS-based perturbation phase. This means that VND failed to empty $\Gamma_{\bar{w}}$, and it is not subsequently applied to another subset of the partition. Instead, a ILS-based perturbation phase is applied to disturb the current solution before another trial of VND.

Figure 1 illustrates searching moves within each neighborhood of VND. The idea in part 1(a) is to move request r from $\Gamma_{\bar{w}}$ to a different subset Γ_i, $i = 1, \ldots, W$, $i \neq \bar{w}$,

as long as r can be routed in G_i. Part 1(b) shows the attempt to accommodate r in $\Gamma_{w'}$ by reallocating as much requests from $\Gamma_{w'}$ as possible. Part 1(c) represents swap moves.



(a) N_1 (b) N_2 (c) N_3

Figure 1: Searching moves within neighborhoods N_1, N_2, and N_3 of VND.

3.2 ILS-based perturbation phase

ILS employs perturbation techniques to escape from a current local optima. Let s^* be a current solution, and let initially s be equal to s^*, then ILS's iterative step has three components: (i) a local search applied to s generating s'; (ii) an acceptance criterion to either update s^* to s' or not; and (iii) a perturbation technique applied to s^* generating a new solution s. The overall best solution is returned after a number of iterations. The reader is referred to Lourenço et al. [16] for a broader discussion on ILS features.

We exploit the idea of applying a perturbation technique, in the manner of ILS, after a solution had been obtained by the VND. Suppose the perturbation is applied when VND failed to swap a request $r \in \Gamma_{\bar{w}}$ within N_3. Note that at this point VND might have successfully performed moves when traversing the requests' list $L_{\bar{w}}$. We do not undo these moves and consequently, when perturbation is applied, the subsets of Γ characterizing the partition may have other elements than those at the beginning of the iteration. The perturbation relies upon an assignment problem built to rearrange requests among the subsets partitioning $\Gamma - \Gamma_{\bar{w}}$. We pull out a randomly chosen request r_i from each subset Γ_i, $\Gamma_i \subseteq \Gamma - \Gamma_{\bar{w}}$, of the partition. Let $G_i^a = (V, E_i^a)$ be the subgraph of G where $E_i^a = E_i \cup F_{r_i}$ (the arcs used to route request r_i are reactivated in G_i^a). The perturbation is done by finding the best way to assign $W-1$ requests chosen to $W-1$ subsets partitioning $\Gamma - \Gamma_{\bar{w}}$. The cost of assigning request $r_i \in \Gamma_i$ to subset Γ_j is given as follows:

- $c_{r_i j} = 2$, if $j = i$;

- $c_{r_i j} = \infty$, if there is no path to route request r_i in the graph G^a_j;
- $c_{r_i j} = 1 - \frac{min_{spl}(r_i)}{min_{spl}(r_i)(G^a_j)} - \max(min_{spl}(r_i)(G^a_i) - min_{spl}(r_i)(G^a_j), 0)$, otherwise;

where $min_{spl}(r_i)(G')$ is the number of arcs of the shortest path between s_{r_i} and d_{r_i} in graph G' (we remember that $min_{spl}(r_i)$ is the number of arcs of the shortest path in the original graph G). The assignment problem results in a feasible partition of $\Gamma - \Gamma_{\bar{w}}$ because a solution that returns each r_i to its original subset Γ_i is always feasible and possess a cost of $2(W-1)$, which is smaller than the cost of any solution that assigns a request r_i to a subset Γ_j where there is no path to route r_i in G^a_j. If the solution of the assignment problem has cost smaller than $2(W-1)$, then at least two requests were assigned to subsets other than their original ones while preserving feasibility in terms of arc disjoint paths to route requests of each subset. Thus, after solving the assignment problem, we check whether the allocation of requests to partition subsets has changed. If so, the algorithm resumes the VND by trying a move with $r \in \Gamma_{\bar{w}}$ within N_1. Otherwise, a perturbation step is done again. Figure 2 shows an example where Γ is partitioned in five subsets. Suppose VND failed to swap a request $r \in \Gamma_4$ within N_3. Then, for each i, $i \in \{1, 2, 3, 5\}$, a request r_i is randomly chosen and the assignment problem built. Note the graph G_4 ($\bar{w} = 4$) does not take part on the assignment.

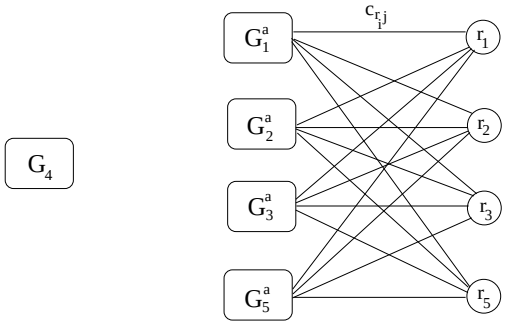


Figure 2: ILS-based perturbation by means of an assignment problem.

We propose two variants for choosing a request r_i from subset Γ_i to build the assignment problem used to guide the perturbation. In the first one, denoted ILS$_p$, all requests belonging to Γ_i have the same probability to be chosen. In the second one, denoted ILS$_{5p}$, requests whose paths pass through the origin s_{r_i} or the destination d_{r_i} of the request $r \in \Gamma_{\bar{w}}$ for which VND failed to perform a move within N_3 have higher probability of being chosen. In fact, if the path used to route request $r_i \in \Gamma_i$ in G has an outgoing (resp. incoming) arc from s_{r_i} (resp. to d_{r_i}), then r_i has five times more probability of being chosen.

4 Computational results

The computational experiments were structured into three comparative settings: comparison among combinations of VND-ILS strategies, evaluation of the effectiveness of the VND-ILS with respect to a multistart heuristic and robustness to initial solution, and comparison with the strongest metaheuristic in the literature. For such purpose, we used benchmark instances from the literature - the most studied realistic instances, and sets Y and Z introduced by Noronha et al. [20]. The realistic instances are available on the web[1] or were provided along with sets Y and Z by Noronha[2]. Instances Y and Z are the most difficult ones. Set Y is formed by randomly generated instances with 100 nodes and different values for the probabilities of a link and a request between a pair of nodes. Set Z is formed by instances with 100 nodes on a grid embedded on a torus where each node is connected only to its nearest four nodes, and different values for the probability of a request between a pair of nodes. Computational experiments were carried out on a Core 2 Duo with 1.97 GHz and 4 GB of RAM, running MS Windows XP, and the proposed algorithms were coded on C++.

Table 1 presents average results to decide among VND-ILS alternatives the one that seems to work better. The first column lists the group of instances. The group Realistic has 26 instances, and each of the Z and Y groups has 5 and 25 instances, respectively. Then, for each combination VND-ILS we report average deviation gaps for 5 runs of the heuristic on each instance, each run limited to 5 minutes. The gaps are calculated as the difference in percentage between the upper bound UB obtained by the heuristic and the lower bound LB computed according to Jaumard et al. [9, 11] and also used in the study conducted by Noronha et al. [20, 21], i.e., (UB-LB)/LB. It can be seen that if the VND strategy is fixed, ILS_{5p} leads consistently to better results. It shows that giving more chance, in the ILS-based perturbation, to requests passing through the origin or destination nodes of the request blocked during the precedent VND phase is more effective in rearranging wavelengths to another VND trial. The different VND strategies have similar behavior, with VND_e obtaining slightly better results. Thus, we present in the sequel results comparing the VND_e-ILS_{5p} with the powerful methods in the literature.

The second comparison setting aims to see whether VND and ILS are effective to improve results regarding a multistart variant of what is considered the best constructive heuristic in the literature, and on the other hand evaluate the robustness of the method with respect to initial solution. The initial solution for VND_e-ILS_{5p} is given by one run of the BFD heuristic proposed by Skorin-Kapov [25]. Thus, we compare VND_e-ILS_{5p} to a multistart variant of BFD where requests whose shortest

[1] http://dag.cs.uni-kl.de/research/rwa/

[2] personal communication

paths in G have the same length are randomly ordered. We then apply solely VND$_e$ to the initial solution. Finally, we apply VND$_e$-ILS$_{5p}$ starting from a different initial solution which is obtained with the edge disjoint path (EDP) heuristic proposed by Manohar et al. [17].

Table 2 presents results of the four methods - Multistart-BDF, BFD-VND$_e$, VND$_e$-ILS$_{5p}$, EDP-VND$_e$-ILS$_{5p}$ - on a set of realistic instances. For each method we performed again 5 runs on each instance, each of them limited to 5 minutes. The first column presents the instance's identification, and from the second to the fourth column the corresponding number of nodes, number of arcs, and number of lightpath requests. For the instances marked with (*) we randomly generated asymmetric requests. Then, in the next columns, we report for each method the minimum number of wavelengths obtained on 5 runs, that is the best solution found, and the average gap. The last column presents the lower bounds. It has seen in Table 1 that realistic instances are easier to solve than the ones in sets Y and Z. Nevertheless, Multistart-BFD and BFD-VND$_e$ were not able to find optimal solutions for all of them, whereas VND$_e$-ILS$_{5p}$ and EDP-VND$_e$-ILS$_{5p}$ did. We see from Table 2 that BFD-VND$_e$ improved results upon Multistart-BFD reducing the overall average gap from 3.27% to 2.50%, though BFD-VND$_e$ still shows some high average gaps of 20% and 15.37%. But when ILS$_{5p}$ was coupled, the method VND$_e$-ILS$_{5p}$ closed all gaps independently of the initial solution.

Tables 3 to 5 present results on the set Y of instances [20, 21]. These 100 node instances are characterized by the probabilities that there is a pair of direct arcs between a pair of nodes, which are 0.03, 0.04, 0.05, and that there is a request between a pair of nodes, which are 0.2, 0.4, 0.6, 0.8, 1. There are 5 instances for each combination of probabilities. For example, instance y.3.60.4, has probability of 0.03 for arcs and 0.6 for requests, and it is the 4th instance with this combination. Results are reported analogously to those in Table 2, and again 5 runs limited to 5 minutes were performed for each instance. Comparing the four methods, they follow the general tendency observed in Table 2. The VND solely produces only slightly reductions on average gaps, and both versions employing ILS-based perturbation lead to significant smaller gaps than Multistart-BFD. We note that VND$_e$-ILS$_{5p}$ was able to improve gaps on all the 42 out of 75 instances for which Multistart-BFD was not able to find the optimum. Moreover, EDP-VND$_e$-ILS$_{5p}$, starting from a different initial solution, was able to obtain smaller gaps than Multistart-BFD on 35 out of these 42 instances. The use of BFD as initial solution yields however better results, since for these harder instances VND$_e$-ILS$_{5p}$ clearly outperforms EDP-VND$_e$-ILS$_{5p}$.

Table 6 presents results on the set Z of instances [20, 21]. These instances have approximately 100 nodes which are the vertices on grids of dimensions 10×10, 8×13, 6×7, 5×20, 4×25, with probability of 0.2, 0.4, 0.6, 0.8, 1 to have a request between

a pair of nodes. For example, instance z.8×13.80, is a 8×13 grid with probability of 0.8 for requests. Results for 5 runs limited to 5 minutes for each method are reported in the same manner. Although gaps of 20% are not observed as in some cases of set Y, these instances seem to be more difficult because Multistart-BFD were not able to match lower bounds. And in this case VND$_e$-ILS$_{5p}$ was able to obtain better gaps than Multistart-BFD for all 25 instances in set Z. Besides, VND$_e$-ILS$_{5p}$ was able to find the optimum for 6 instances. As it can be seen from Table 6, the proposed approach of combining VND and ILS-based strategies is robust. EDP-VND$_e$-ILS$_{5p}$, even though not using BFD to generate initial solutions, found better gaps than Multistart-BFD for 14 instances.

The third setting is intended to compare VND$_e$-ILS$_{5p}$ with the genetic algorithm GA-RWA proposed by Noronha et al. [21] which recently improved the state-of-the-art algorithms in the literature. Noronha et al. [21] selected a subset of 30 instances as the hardest ones, and their results showed that GA-RWA was able to improve average gaps on 29 of them. Table 7 presents a comparison between VND$_e$-ILS$_{5p}$ and GA-RWA on these 30 instances. Results of GA-RWA were obtained with a limit of 10 minutes on a Pentium IV with 3.4GHz, and those of VND$_e$-ILS$_{5p}$ with a limit of 5 minutes as an attempt to take into account the processor difference. The first column presents the instance. Then, the second and third (resp. the fourth and fifth) columns present the best solution and the average gap for 5 runs obtained by GA-RWA (resp. by VND$_e$-ILS$_{5p}$). The last column presents the lower bound. Table 7 shows quite competitive results regarding the state-of-the-art in the literature. VND$_e$-ILS$_{5p}$ found better solutions than GA-RWA in 26 out of 30 instances, and equal solutions for the other 4. Besides, VND$_e$-ILS$_{5p}$ was able to find 6 optima for instances that were still open. In terms of time consuming, on average, the time spent on neighborhoods N_1 and N_2 is about 5%, on neighborhood N_3 is about 25%, and the most time consuming phase is the ILS-based perturbation with 70% of the CPU time.

We now proceed to detailed analysis on the behavior of algorithm VND$_e$-ILS$_{5p}$. The first analysis is an attempt to learn more about the performance of VND$_e$-ILS$_{5p}$ if it should run on less time than 5 minutes. For such purpose, we make use of the time-to-target (TTT) plots proposed by Aiex et al. [1, 2]. The hypothesis behind is that CPU times fit a two parameter exponential distribution. Thus, for a given problem instance, TTT plots display on the ordinate axis the probability for the heuristic to obtain a solution as good as a given target value within a running time in seconds, shown on the abscissa axis. We had chosen 8 instances from Table 7 to do this. For each instance we ran VND$_e$-ILS$_{5p}$ 200 times with different seeds until the algorithm reaches the target value. Figures 3 to 6 show the TTT plots generated with 4 instances for which an average gap of 0% is reached - ATT, NSF.12, y.4.20.4, y.5.100.2. For these instances we set the target value to the optima. On the other hand, Figures 7

to 10 show the TTT plots generated with the instances for which a high average gap is observed - y.4.80.1, y.4.100.1, Z.10x10.60, Z.10x10.80. In these cases, we set the target value to one wavelength less than the best solution found by GA-RWA. TTT plots indicate that VND$_e$-ILS$_{5p}$ is likely to obtain high quality solutions in a short time. As it can be seen, on instances tested, VND$_e$-ILS$_{5p}$ has high probability to find very good solutions (optimum or better than the best known) in less than 2 minutes, except for instance y.5.100.2, for which a little more time is needed.

The second analysis tries to bring some insight on which neighborhood brings the best gain. We performed experiments with reduced versions of VND using, along with ILS-based perturbation, neighborhoods N_1 and N_2 separately and in combination with N_3. Neighborhood N_3 is not tested separately since alone it cannot reduce the number of wavelengths. Table 8 shows numerical results for the reduced versions of VND. The first column presents the instance. Then, the subsequent pairs of columns present the best solution and the average gap for each reduced version of VND. As before, it is reported for each proposed algorithm the results of 5 runs with a time limit of 5 minutes each. The last column presents the lower bound. We remark that although N_3 alone cannot be used to improve solutions, the reduced versions of VND using it in combination with N_1 or N_2 lead to the better results. So, a move in neighborhood N_3 is an important instrument to rearrange requests to further improving moves. It is also worth noting that neighborhood N_2 alone is not effective, but in combination with N_3 becomes the most successful reduced version of VND.

5 Conclusions

We propose an algorithm for the RWA problem that alternates between a VND phase and a ILS-based perturbation phase. In the VND phase we explore three neighborhoods by trying three kinds of moves. The purpose is to reduce one wavelength. The ILS-based phase is called to introduce a perturbation in the current partition of the lightpath requests' set. The perturbation itself does not improve a solution, but it leaves a rearrangement of lightpaths among subsets of the partition that may yield further improvements with another trial of VND. Even though better results were found taking BFD as constructive heuristic, VND-ILS is quite robust with respect to the initial solution and clearly outperforms a multistart variant of BFD. Computation experiments were conducted on the hardest benchmark instances, and significant improvements upon better upper bounds from the literature were achieved. An interesting research direction is to adapt the methods to deal with possibilities of traffic-grooming.

Acknowledgements

The authors wish to thank the two anonymous referees for helpful suggestions in improving this paper. The authors from Brazilian institutions were partially supported by CAPES, CNPq, and FAPEMIG, Brazil.

References

[1] R.M. Aiex, M.G.C. Resende and C.C. Ribeiro, "Probability distribution of solution time in GRASP: an experimental investigation", *Journal of Heuristics* 8 (2002), 343–373.

[2] R.M. Aiex, M.G.C. Resende and C.C. Ribeiro, "TTT plots: a perl program to create time-to-target plots", *Optimization Letters* 1 (2007), 355–366.

[3] D. Banerjee and B. Mukherjee, "A practical approach for routing and wavelength assignment in large wavelength-routed optical networks", *IEEE Journal on Selected Areas in Communications* 14 (1996), 903–908.

[4] I. Chlamtac, A. Ganz, and G. Karmi,"Lightpath communications: An approach to high bandwidth optical WAN's", *IEEE Transactions on Communications* 40 (1992), 1171–1182.

[5] R. Dutta and G.N. Rouskas, "A survey of virtual topology design algorithms for wavelength routed optical networks", *Optical Networks Magazine* 1 (2000), 73–89.

[6] P. Hansen, N. Mladenović "Variable neighborhood search", **In:** F. Glover, G. Kochenberger (Eds.), *Handbook of Metaheuristics*, Kluwer, 145–184, 2003.

[7] P. Hansen, N. Mladenović, J.A. Moreno Pérez, "Variable neighborhood search: methods and applications", *4OR A Quarterly Journal of Operations Research* 6 (2008), 319–360.

[8] P. Hansen, N. Mladenović , J.A. Moreno Pérez, "Variable neighbourhood search: methods and applications", *Annals of Operations Research* 175 (2010), 367–407.

[9] B. Jaumard, C. Meyer, B. Thiongane, "ILP formulations for the RWA problem for symmetrical systems", **In:** P. Pardalos, M. Resende (Eds.), *Handbook for Optimization in Telecommunications*, Kluwer, 637–678, 2006.

[10] B. Jaumard, C. Meyer, B. Thiongane, "Comparison of ILP formulations for the RWA problem", *Optical Switching and Networking* 4 (2007), 157–172.

[11] B. Jaumard, C. Meyer, B. Thiongane, "On column generation formulations for the RWA problem", *Discrete Applied Mathematics* 157 (2009), 1291–1308.

[12] J. Kennington, E. Olinick, A. Ortynski, and G. Spiride, "Wavelength routing and assignment in a survivable WDM mesh network", *Operations Research* 51 (2003), 67–79.

[13] R.M. Krishnaswamy and K.N. Sivarajan, "Algorithms for routing and wavelength assignment based on solutions of LP-relaxations", *IEEE Communications Letters* 5 (2001), 435–437.

[14] R.M. Krishnaswamy and K.N. Sivarajan, "Design of logical topologies: A linear formulation for wavelength-routed optical networks with no wavelength changers", *IEEE/ACM Transactions on Networking* 9 (2001), 186–198.

[15] K. Lee, K.C. Kang, T. Lee, S. Park, "An optimization approach to routing and wavelength assignment in WDM all-optical mesh networks without wavelength conversion", *ETRI Journal* 24 (2002), 131–141.

[16] H.R. Lourenço, O.C. Martin, T. Stützle, "Iterated local search", **In:** P. Glover, G. Kochenberger (Eds.), *Hanbook of Metaheuristics*, 321–353, Springer, 2003.

[17] P. Manohar, D. Manjunath, R.K. Shevgaonkar, "Routing and wavelengths assignment in optical networks from edge disjoint path algorithms", *IEEE Communication Letters* 6 (2002), 211–213.

[18] N. Mladenović, P. Hansen, "Variable neighbourhood search", *Computers & Operations Research* 24 (1997), 1097–1100.

[19] B. Mukherjee, D. Banerjee, S. Ramamurthy, and A. Mukherjee, "Some principles for designing a wide-area WDM optical network", *IEEE/ACM Transactions on Networking* 4 (1996), 684–696.

[20] T. Noronha, M.G.C. Resende, C.C. Ribeiro, "Efficient implementations of heuristics for routing and wavelength assignment", **In:** C.C. McGeoch (Ed.), Proceedings of the 7th International Workshop on Experimental Algorithms, *Lecture Notes in Computer Science* 5038 (2008), 169–180.

[21] T. Noronha, M.G.C. Resende, C.C. Ribeiro, "A biased random-key genetic algorithm for routing and wavelength assignment", *Journal of Global Optimization*, DOI 10.1007/s10898-010-9608-7.

[22] T. Noronha, C.C. Ribeiro, "Routing and wavelength assignment by partition colouring", *European Journal of Operational Research* 171 (2006), 797–810.

[23] R. Ramaswami and K.N. Sivarajan, "Routing and wavelength assignment in all-optical networks", *IEEE/ACM Transactions on Networking* 3 (1995), 489–500.

[24] R. Ramaswami and K.N. Sivarajan, "Design of logical topologies for wavelength-routed optical networks", *IEEE Journal on Selected Areas in Communications* 14 (1996), 840–851.

[25] N. Skorin-Kapov,"Routing and wavelength assignment in optical networks using bin packing based algorithms", *European Journal of Operational Research* 177 (2007), 1167–1179.

[26] H. Zang, J.P. Jue, and B. Mukherjee, "A review of routing and wavelength assignment approaches for wavelength-routed optical WDM networks", *Optical Networks Magazine* 1 (2000), 47–60.

Set	VND_e-ILS_p	VND_r-ILS_p	VND_e-ILS_{5p}	VND_r-ILS_{5p}
Realistic	0.00	0.00	0.00	0.00
Z.20	2.72	2.57	2.36	2.39
Z.40	3.27	3.15	2.75	2.78
Z.60	3.11	3.08	2.55	2.62
Z.80	2.66	2.63	2.41	2.40
Z.100	4.15	4.15	3.94	3.88
Y.3	4.25	4.20	3.66	3.75
Y.4	6.62	6.60	6.37	6.39
Y.5	1.79	1.66	1.30	1.33
Average	3.17	3.12	2.82	2.84

Table 1: Deviation gaps in percentage for combinations of VND-ILS strategies.

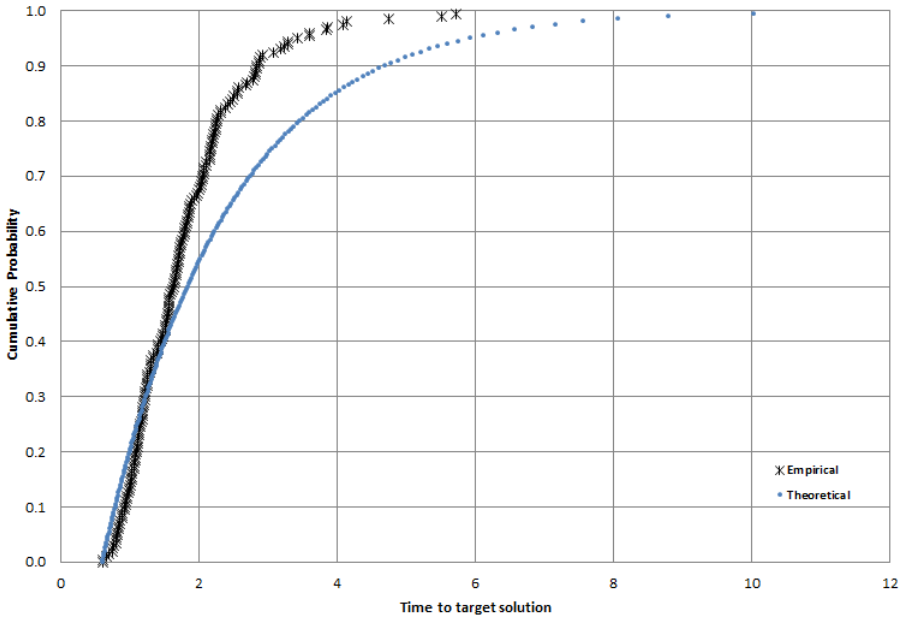


Figure 3: TTT plots produced for instance ATT - target value equal to 20.

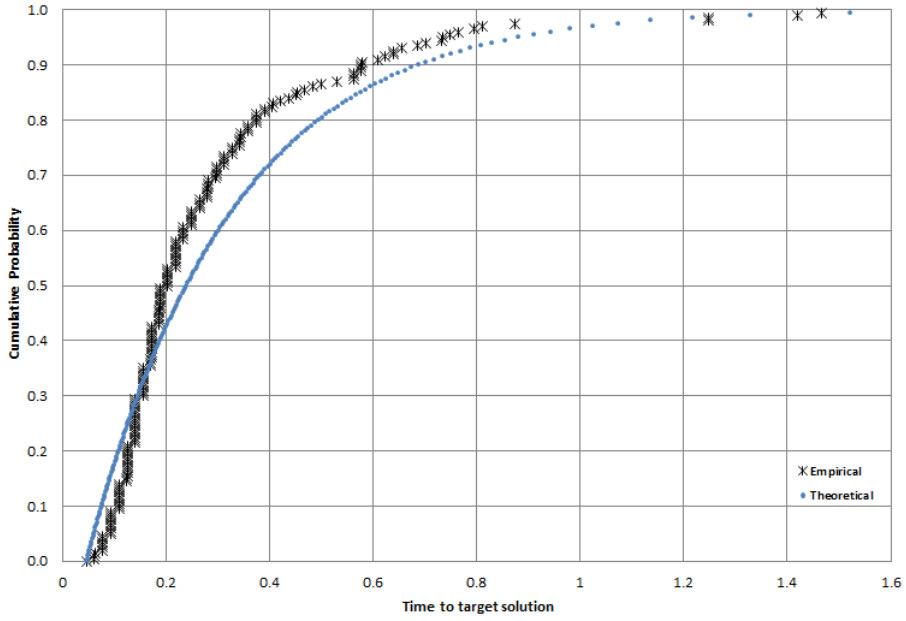


Figure 4: TTT plots produced for instance NSF.12 - target value equal to 38.

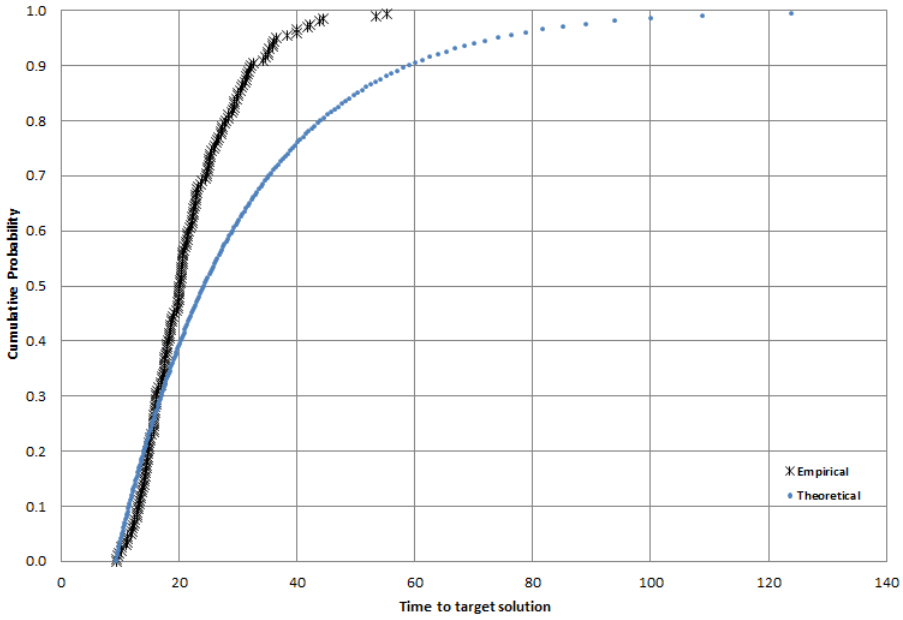


Figure 5: TTT plots produced for instance y.4.20.4 - target value equal to 19.

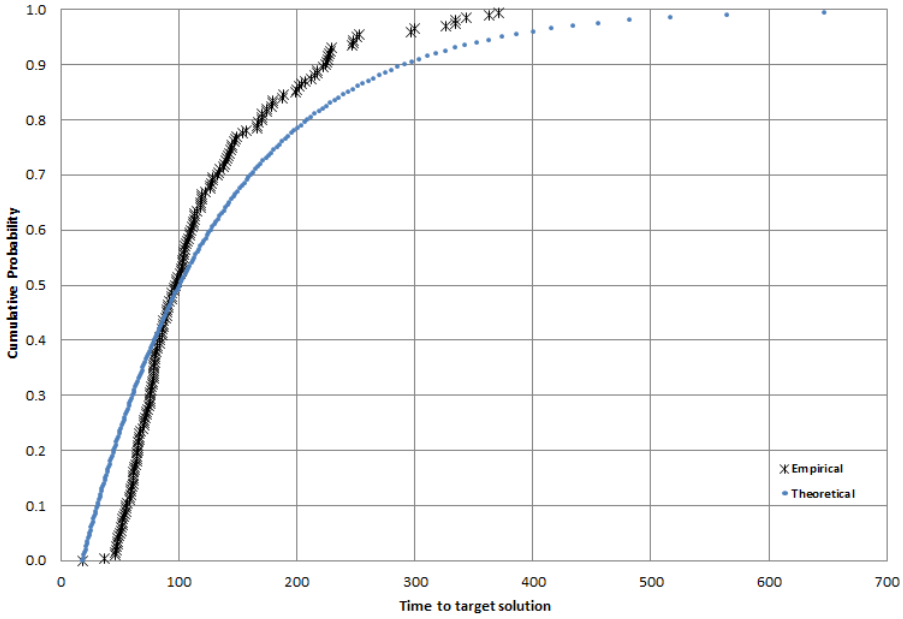


Figure 6: TTT plots produced for instance y.5.100.2 - target value equal to 73.

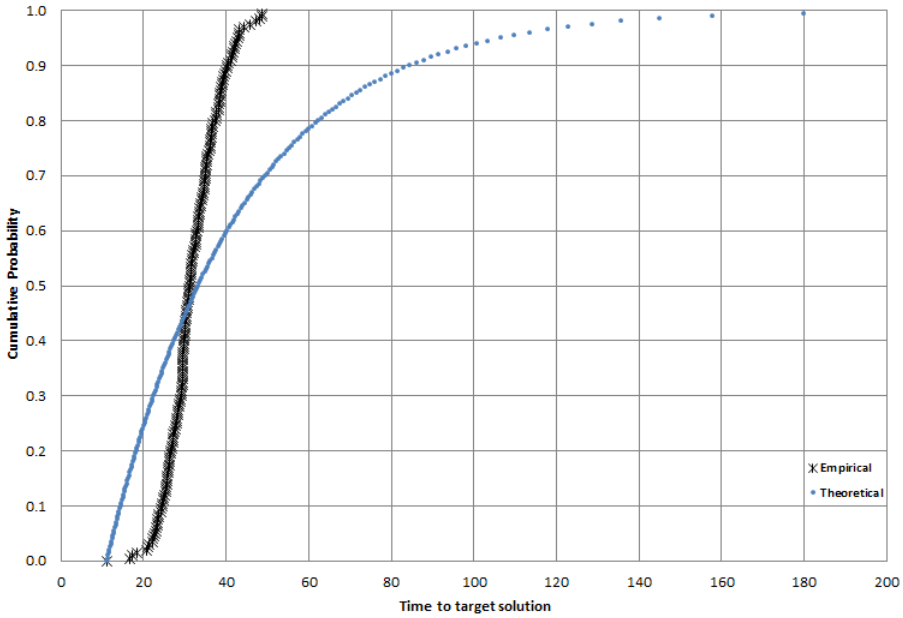


Figure 7: TTT plots produced for instance y.4.80.1 - target value equal to 72.

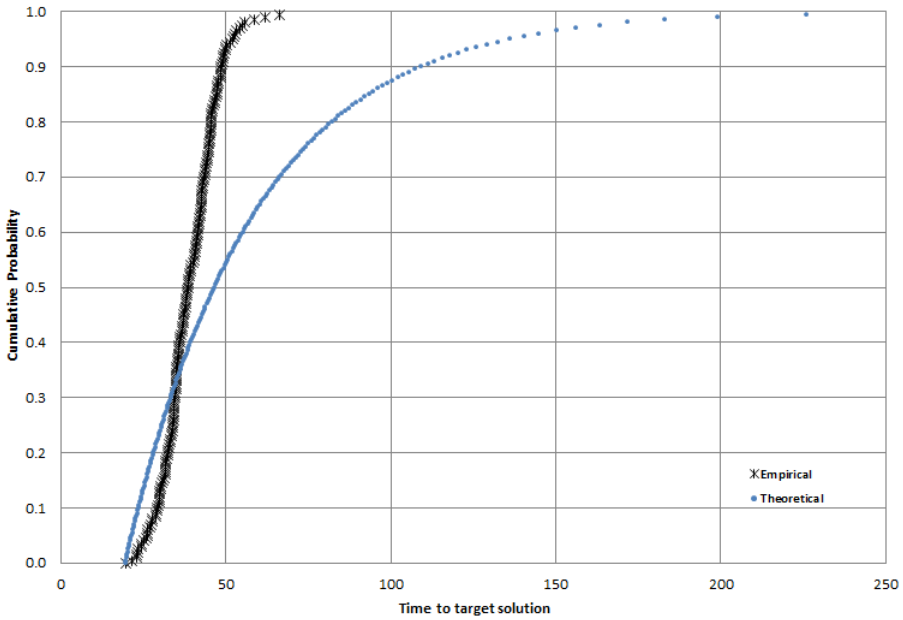


Figure 8: TTT plots produced for instance y.4.100.1 - target value equal to 89.

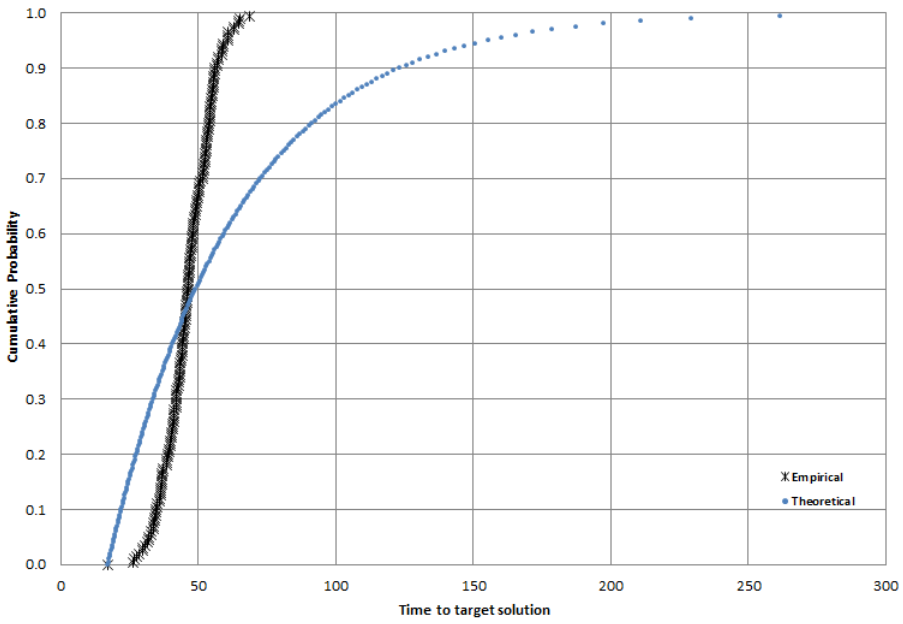


Figure 9: TTT plots produced for instance Z.10x10.60 - target value equal to 86.

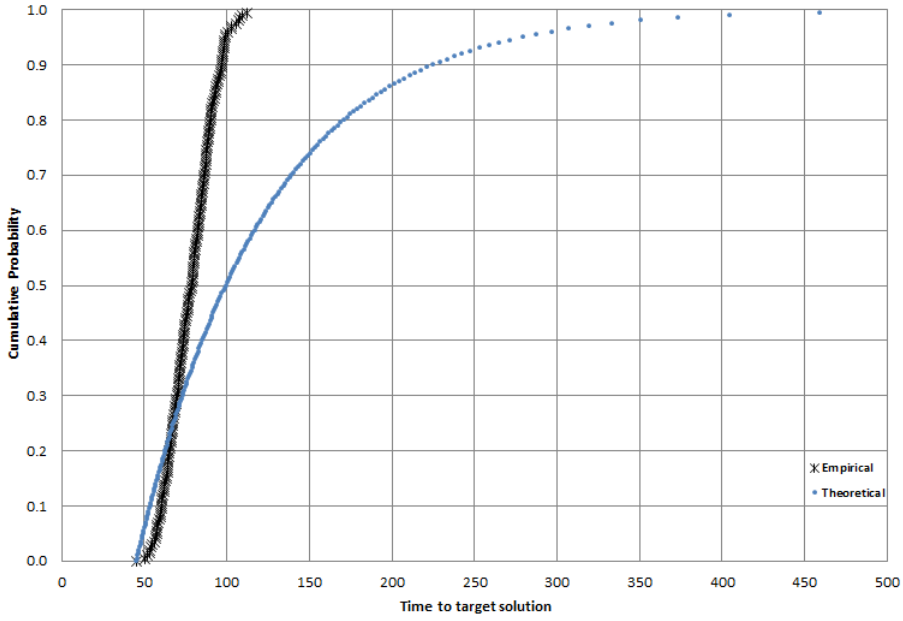


Figure 10: TTT plots produced for instance Z.10x10.80 - target value equal to 114.

				Multistart-BFD		BFD-VND$_e$		VND$_e$-ILS$_{5p}$		EDP-VND$_e$-ILS$_{5p}$		
Instance	$\|V\|$	$\|E\|$	$\|\Gamma\|$	λ_{min}	gap(%)	λ_{min}	gap(%)	λ_{min}	gap(%)	λ_{min}	gap(%)	LB
Atlanta20	15	22	13680	1342	6.91	1284	2.37	1256	0.00	1256	0.00	1256
ATT	90	137	359	25	25.00	24	20.00	20	0.00	20	0.00	20
ATT2	71	175	2918	113	0.71	113	0.71	113	0.00	113	0.00	113
Brasil	27	70	1370	48	0.00	48	0.00	48	0.00	48	0.00	48
Cost266*	37	57	6543	446	0.00	446	0.00	446	0.00	446	0.00	446
Dfn-bwin*	10	45	4840	73	0.00	73	0.00	73	0.00	73	0.00	73
Dfn-gwin*	11	47	3771	316	0.00	316	0.00	316	0.00	316	0.00	316
EON	20	39	373	22	0.00	22	0.00	22	0.00	22	0.00	22
Finland	31	51	930	47	2.17	47	2.17	46	0.00	46	0.00	46
France*	25	45	15398	946	0.00	946	0.00	946	0.00	946	0.00	946
Germany50	50	88	4730	169	15.37	169	15.37	147	0.00	147	0.00	147
Giul	39	86	14732	402	6.07	401	5.91	379	0.00	379	0.00	379
Janos-us*	26	42	3262	215	4.15	215	4.15	207	0.00	207	0.00	207
Nobel-eu	28	41	3796	304	0.00	304	0.00	304	0.00	304	0.00	304
Nobel-germany	17	26	1320	89	4.94	89	4.71	85	0.00	85	0.00	85
Norway	27	51	10696	543	0.00	543	0.00	543	0.00	543	0.00	543
NSF.1	14	21	284	23	4.55	22	0.00	22	0.00	22	0.00	22
NSF.3	14	21	285	22	3.64	22	2.73	22	0.00	22	0.00	22
NSF.12	14	21	551	39	2.63	39	2.63	38	0.00	38	0.00	38
NSF.48	14	21	547	41	0.49	41	0.00	41	0.00	41	0.00	41
NSF2.1	14	22	284	21	0.00	21	0.00	21	0.00	21	0.00	21
NSF2.3	14	22	285	21	0.00	21	0.00	21	0.00	21	0.00	21
NSF2.12	14	22	551	35	1.71	35	0.00	35	0.00	35	0.00	35
NSF2.48	14	22	547	39	0.00	39	0.00	39	0.00	39	0.00	39
Sun	27	51	952	61	3.39	60	1.69	59	0.00	59	0.00	59
Average					3.27		2.50		0.00		0.00	

Table 2: Comparison with multistart BFD on realistic instances.

	Multistart-BFD		BFD-VND$_e$		VND$_e$-ILS$_{5p}$		EDP-VND$_e$-ILS$_{5p}$		
Instance	λ_{min}	gap(%)	λ_{min}	gap(%)	λ_{min}	gap(%)	λ_{min}	gap(%)	LB
y.3.20.1	33	22.22	33	22.22	29	8.89	30	11.11	27
y.3.20.2	33	0.00	33	0.00	33	0.00	33	0.00	33
y.3.20.3	31	6.90	31	6.90	29	0.00	29	0.00	29
y.3.20.4	31	19.23	31	19.23	28	8.46	29	11.54	26
y.3.20.5	31	10.71	31	10.71	28	2.86	29	3.57	28
y.3.40.1	62	18.49	62	18.11	57	7.55	59	11.32	53
y.3.40.2	59	0.00	59	0.00	59	0.00	59	0.00	59
y.3.40.3	61	0.00	61	0.00	61	0.00	61	0.00	61
y.3.40.4	58	16.00	58	16.00	54	8.00	56	12.00	50
y.3.40.5	60	13.21	60	13.21	56	6.79	58	9.43	53
y.3.60.1	93	14.81	93	14.81	87	7.41	90	12.35	81
y.3.60.2	89	0.00	89	0.00	89	0.00	89	0.00	89
y.3.60.3	91	0.00	91	0.00	91	0.00	91	0.00	91
y.3.60.4	85	9.23	85	8.97	80	2.82	83	6.41	78
y.3.60.5	86	12.73	86	12.73	82	7.53	85	11.69	77
y.3.80.1	123	16.04	123	16.04	115	9.25	122	15.09	106
y.3.80.2	117	0.00	117	0.00	117	0.00	117	0.00	117
y.3.80.3	118	0.17	118	0.17	118	0.00	118	0.00	118
y.3.80.4	112	6.67	111	6.48	106	1.14	110	5.71	105
y.3.80.5	114	9.62	114	9.62	109	4.81	114	9.62	104
y.3.100.1	151	15.27	151	15.57	143	9.31	152	16.03	131
y.3.100.2	146	0.00	146	0.00	146	0.00	146	0.00	146
y.3.100.3	146	0.00	146	0.00	146	0.00	146	0.00	146
y.3.100.4	138	5.80	138	5.80	132	1.22	139	6.87	131
y.3.100.5	141	9.30	141	9.30	136	5.58	143	10.85	129
Average		8.26		8.23		3.66		6.14	

Table 3: Comparison with multistart BFD on set $Y3$ of instances.

	Multistart-BFD		BFD-VND$_e$		VND$_e$-ILS$_{5p}$		EDP-VND$_e$-ILS$_{5p}$		
Instance	λ_{min}	gap(%)	λ_{min}	gap(%)	λ_{min}	gap(%)	λ_{min}	gap(%)	LB
y.4.20.1	21	23.53	21	23.53	19	11.76	19	11.76	17
y.4.20.2	28	0.00	28	0.00	28	0.00	28	0.00	28
y.4.20.3	23	0.00	23	0.00	23	0.00	23	0.00	23
y.4.20.4	20	8.42	20	7.37	19	0.00	19	0.00	19
y.4.20.5	21	23.53	21	23.53	19	11.76	19	15.29	17
y.4.40.1	38	22.58	38	22.58	35	12.90	36	16.13	31
y.4.40.2	57	0.00	57	0.00	57	0.00	57	0.00	57
y.4.40.3	43	0.00	43	0.00	43	0.00	43	0.00	43
y.4.40.4	38	0.00	38	0.00	38	0.00	38	0.00	38
y.4.40.5	40	21.21	40	21.21	37	12.12	38	15.15	33
y.4.60.1	56	19.15	56	19.15	53	12.77	54	14.89	47
y.4.60.2	86	0.00	86	0.00	86	0.00	86	0.00	86
y.4.60.3	64	0.00	64	0.00	64	0.00	64	0.00	64
y.4.60.4	58	0.00	58	0.00	58	0.00	58	0.00	58
y.4.60.5	58	19.18	58	18.78	55	12.24	57	16.33	49
y.4.80.1	73	57.02	73	56.60	70	48.94	72	54.04	47
y.4.80.2	118	0.00	118	0.00	118	0.00	118	0.00	118
y.4.80.3	81	0.00	81	0.00	81	0.00	81	0.00	81
y.4.80.4	78	0.00	78	0.00	78	0.00	78	0.00	78
y.4.80.5	76	16.92	76	16.92	72	10.77	75	15.38	65
y.4.100.1	91	19.74	91	19.74	86	14.21	90	19.47	76
y.4.100.2	146	0.00	146	0.00	146	0.00	146	0.00	146
y.4.100.3	98	0.00	98	0.00	98	0.00	98	0.00	98
y.4.100.4	98	0.00	98	0.00	98	0.00	98	0.00	98
y.4.100.5	93	16.50	93	16.50	89	11.75	93	17.25	80
Average		9.91		9.84		6.37		7.83	

Table 4: Comparison with multistart BFD on set $Y4$ of instances.

	Multistart-BFD		BFD-VND$_e$		VND$_e$-ILS$_{5p}$		EDP-VND$_e$-ILS$_{5p}$		
Instance	λ_{min}	gap(%)	λ_{min}	gap(%)	λ_{min}	gap(%)	λ_{min}	gap(%)	LB
y.5.20.1	14	7.69	14	7.69	13	0.00	13	0.00	13
y.5.20.2	17	0.00	17	0.00	17	0.00	17	0.00	17
y.5.20.3	13	8.33	13	8.33	12	5.00	13	8.33	12
y.5.20.4	17	0.00	17	0.00	17	0.00	17	0.00	17
y.5.20.5	15	0.00	15	0.00	15	0.00	15	0.00	15
y.5.40.1	25	4.17	25	4.17	24	0.00	24	0.83	24
y.5.40.2	31	1.29	31	1.29	31	0.00	31	0.00	31
y.5.40.3	24	9.09	24	9.09	23	4.55	23	6.36	22
y.5.40.4	33	0.00	33	0.00	33	0.00	33	0.00	33
y.5.40.5	28	0.00	28	0.00	28	0.00	28	0.00	28
y.5.60.1	36	11.52	36	11.52	35	6.67	36	9.09	33
y.5.60.2	45	1.33	45	0.89	45	0.00	45	0.00	45
y.5.60.3	35	2.94	35	2.94	34	0.00	34	0.00	34
y.5.60.4	48	0.00	48	0.00	48	0.00	48	0.00	48
y.5.60.5	40	0.00	40	0.00	40	0.00	40	0.00	40
y.5.80.1	47	11.16	47	11.16	46	6.98	47	9.30	43
y.5.80.2	60	1.69	60	1.69	59	0.00	59	1.02	59
y.5.80.3	45	4.65	45	4.65	44	2.33	45	5.12	43
y.5.80.4	63	0.00	63	0.00	63	0.00	63	0.00	63
y.5.80.5	53	0.00	53	0.00	53	0.00	53	0.00	53
y.5.100.1	59	7.27	59	7.27	57	3.64	58	5.82	55
y.5.100.2	75	2.74	75	2.74	73	0.00	74	1.37	73
y.5.100.3	56	5.66	56	5.66	54	3.40	56	5.66	53
y.5.100.4	77	0.00	77	0.00	77	0.00	77	0.00	77
y.5.100.5	66	0.00	66	0.00	66	0.00	66	0.00	66
Average		3.18		3.16		1.30		2.12	

Table 5: Comparison with multistart BFD on set $Y5$ of instances.

	Multistart-BFD		BFD-VND$_e$		VND$_e$-ILS$_{5p}$		EDP-VND$_e$-ILS$_{5p}$		
Instance	λ_{min}	gap(%)	λ_{min}	gap(%)	λ_{min}	gap(%)	λ_{min}	gap(%)	LB
Z.10x10.20	31	17.04	31	17.04	29	7.41	30	11.11	27
Z.8x13.20	35	8.48	35	8.48	34	3.03	34	3.03	33
Z.6x17.20	46	4.55	46	4.55	44	1.36	45	2.27	44
Z.5x20.20	55	1.85	55	1.85	54	0.00	54	0.00	54
Z.4x25.20	68	3.03	68	3.03	66	0.00	67	2.12	66
Z.10x10.40	59	15.69	58	14.51	55	8.63	57	12.16	51
Z.8x13.40	67	6.35	67	6.35	64	2.54	66	4.76	63
Z.6x17.40	87	4.29	87	3.57	85	1.19	86	2.38	84
Z.5x20.40	104	2.97	104	2.97	101	0.59	103	2.18	101
Z.4x25.40	129	2.70	129	2.38	127	0.79	129	2.38	126
Z.10x10.60	88	14.29	87	14.03	84	9.09	87	12.99	77
Z.8x13.60	101	5.63	101	5.21	98	2.08	100	4.38	96
Z.6x17.60	133	3.91	133	3.91	129	0.78	131	2.66	128
Z.5x20.60	158	2.60	158	2.60	154	0.26	158	2.60	154
Z.4x25.60	195	2.08	195	1.98	193	0.52	197	2.71	192
Z.10x10.80	116	12.62	116	12.62	112	9.51	117	13.98	103
Z.8x13.80	134	4.50	134	4.50	130	1.09	134	3.88	129
Z.6x17.80	176	3.39	176	3.27	171	0.47	175	2.57	171
Z.5x20.80	209	1.95	209	1.95	206	0.49	210	2.63	205
Z.4x25.80	261	1.56	261	1.56	258	0.47	264	2.72	257
Z.10x10.100	142	13.60	142	13.60	139	11.52	146	17.44	125
Z.8x13.100	175	4.17	175	4.17	173	3.21	178	6.31	168
Z.6x17.100	222	3.06	222	3.06	220	1.85	225	4.17	216
Z.5x20.100	256	2.56	256	2.48	253	1.52	260	4.16	250
Z.4x25.100	319	2.24	318	2.18	317	1.60	325	4.42	312
Average		5.80		5.67		2.80		5.20	

Table 6: Comparison with multistart BFD on set Z of instances.

	GA-RWA		VND_e-ILS_{5p}		
Instance	λ_{min}	gap(%)	λ_{min}	gap(%)	LB
ATT	24	20.0	20	0.0	20
ATT2	113	0.0	113	0.0	113
Finland	46	0.4	46	0.0	46
NSF.3	22	0.9	22	0.0	22
NSF.12	39	2.6	38	0.0	38
NSF2.12	35	0.6	35	0.0	35
Z.10x10.20	31	15.6	29	7.4	27
Z.6x17.40	87	4.0	85	1.2	84
Z.10x10.60	87	13.2	84	9.1	77
Z.4x25.60	195	2.0	193	0.5	192
Z.10x10.80	115	12.4	112	9.5	103
Z.8x13.80	134	3.9	130	1.1	129
Z.6x17.80	176	3.0	171	0.5	171
Z.5x20.80	209	2.0	206	0.5	205
Z.4x25.80	260	1.3	258	0.5	257
Z.5x20.100	257	2.8	253	1.5	250
y.4.20.4	20	6.3	19	0.0	19
y.3.40.5	59	12.8	56	7.4	53
y.3.60.5	86	12.5	82	7.5	77
y.4.60.5	58	18.4	55	12.2	49
y.5.60.1	36	9.7	35	6.7	33
y.3.80.1	122	15.5	115	9.3	106
y.3.80.5	113	8.8	109	4.8	104
y.4.80.1	73	55.3	70	48.9	47
y.4.80.5	75	16.0	72	10.8	65
y.5.80.1	47	11.2	46	7.0	43
y.5.80.2	60	1.7	59	0.0	59
y.4.100.1	90	18.4	86	14.2	76
y.5.100.1	58	5.5	57	3.6	55
y.5.100.2	74	1.6	73	0.0	73
Average		9.3		5.5	

Table 7: Comparison with GA-RWA on a set of difficult instances.

	N_1		N_2		$N_1 + N_2$		$N_1 + N_3$		$N_2 + N_3$		
Instance	λ_{min}	gap(%)	λ_{min}	gap(%)	λ_{min}	gap(%)	λ_{min}	gap(%)	λ_{min}	gap(%)	LI
ATT	20	0.0	26	33.3	20	0.0	20	0.0	20	0.0	20
ATT2	113	0.0	115	2.0	113	0.0	113	0.0	113	0.0	113
Finland	46	0.0	48	5.2	46	0.0	46	0.0	46	0.0	46
NSF.3	22	0.0	22	0.0	22	0.0	22	0.0	22	0.0	22
NSF.12	38	0.0	38	0.0	38	0.0	38	0.0	38	0.0	38
NSF2.12	35	0.0	35	0.0	35	0.0	35	0.0	35	0.0	35
Z.10x10.20	30	14.1	32	20.7	30	11.1	29	7.4	29	7.4	27
Z.6x17.40	87	3.6	88	5.5	85	2.1	86	2.4	85	1.2	84
Z.10x10.60	88	14.3	89	15.8	86	12.7	85	10.4	85	10.4	77
Z.4x25.60	195	2.2	196	3.1	193	1.0	195	1.7	193	0.9	192
Z.10x10.80	115	12.2	116	13.6	115	11.8	114	10.7	113	9.7	103
Z.8x13.80	134	3.9	135	5.0	132	2.9	131	1.9	131	1.6	129
Z.6x17.80	175	2.7	177	4.0	173	1.3	173	1.2	172	0.7	171
Z.5x20.80	209	2.1	210	2.8	206	0.9	207	1.4	206	0.6	205
Z.4x25.80	260	1.5	262	2.2	259	0.8	260	1.3	259	0.8	257
Z.5x20.100	256	2.5	257	3.0	254	1.9	255	2.3	254	1.7	250
y.4.20.4	19	0.0	21	11.6	19	4.2	19	0.0	19	0.0	19
y.3.40.5	58	10.9	60	14.7	58	9.4	57	7.5	57	7.5	53
y.3.60.5	85	11.4	87	13.5	84	10.1	83	8.3	83	7.8	77
y.4.60.5	58	18.4	59	21.2	58	18.4	55	13.5	55	12.2	49
y.5.60.1	36	10.9	37	14.5	36	10.3	35	6.7	35	8.5	33
y.3.80.1	121	14.5	123	17.2	119	12.8	116	10.0	116	9.6	106
y.3.80.5	113	8.7	114	10.0	111	7.5	110	5.8	109	5.4	104
y.4.80.1	73	55.7	74	58.3	73	55.3	70	49.4	70	49.4	47
y.4.80.5	75	15.7	76	17.8	75	15.4	73	12.3	72	11.7	65
y.5.80.1	47	11.2	49	14.0	47	11.2	46	7.0	46	7.9	43
y.5.80.2	59	1.4	60	2.7	59	1.0	59	0.0	59	0.0	59
y.4.100.1	90	18.9	91	20.3	90	18.7	87	14.5	87	14.5	76
y.5.100.1	58	5.5	59	8.4	58	5.5	57	3.6	57	3.6	55
y.5.100.2	74	1.4	76	4.1	74	1.4	73	0.5	73	0.5	73
Average		8.1		11.5		7.6		6.0		5.8	

Table 8: Analysis on the performance of each neighborhood.