J Glob Optim (2016) 65:813–835
DOI 10.1007/s10898-015-0389-x

A biased random-key genetic algorithm to maximize the number of accepted lightpaths in WDM optical networks

Julliany S. Brandão[1] · Thiago F. Noronha[2] · Celso C. Ribeiro[1]

Received: 11 January 2015 / Accepted: 15 November 2015 / Published online: 19 December 2015

Abstract Given a set of lightpath requests, the problem of routing and wavelength (RWA) assignment in wavelength division multiplexing (WDM) optical networks consists in routing a subset of these requests and assigning a wavelength to each of them, such that two lightpaths that share a common link are assigned to different wavelengths. There are many variants of this problem in the literature. We focus in the variant in which the objective is to maximize the number of requests that may be accepted, given a limited set of available wavelengths. This problem is called max-RWA and it is of practical and theoretical interest, because algorithms for this variant can be extended for other RWA problems that arise from the design of WDM optical networks. A number of exact algorithms based on integer programming formulations have been proposed in the literature to solve max-RWA, as well as algorithms to provide upper bounds to the optimal solution value. However, the algorithms based on the state-of-the-art formulations in the literature cannot solve the largest instances to optimality. For these instances, only upper bounds to the value of the optimal solutions are known. The literature on heuristics for max-RWA is short and focus mainly on solving small size instances with up to 27 nodes. In this paper, we propose new greedy constructive heuristics and a biased random-key genetic algorithm, based on the best of the proposed

This work was partially supported by the Brazilian National Council for Scientific and Technological Development (CNPq), the Foundation for Support of Research of the State of Minas Gerais (FAPEMIG), the Foundation for Support of Research of the State of Rio de Janeiro (FAPERJ), and the Coordination for the Improvement of Higher Education Personnel, Brazil (CAPES).

✉ Celso C. Ribeiro
celso@ic.uff.br

Julliany S. Brandão
jbrandao@ic.uff.br

Thiago F. Noronha
tfn@dcc.ufmg.br

1 Universidade Federal Fluminense (UFF), Rua Passo da Pátria 156, Niterói, RJ 24210-240, Brazil

2 Universidade Federal de Minas Gerais (UFMG), Avenida Antônio Carlos 6627, Belo Horizonte, MG 31270-901, Brazil

greedy heuristics. Computational experiments showed that the new heuristic outperforms the best ones in literature. Furthermore, for the largest instances in the literature where only upper bounds to the value of the optimal solutions are known, the average optimality gap of the best of the proposed heuristics is smaller than 4 %.

Keywords Random-key genetic algorithms · Metaheuristics · Optical networks · Routing and wavelength assignment

1 Introduction

In optical networks, information is transmitted as optical signals through optical fibers. Each optical link operates at a speed of the order of terabits per second, which is much faster than the currently available optical devices. *Wavelength Division Multiplexing* (WDM) allows the more efficient use of the capacity of the optical fibers, as far as it permits the simultaneous transmission of different signals along the same fiber, provided they are multiplexed with different wavelengths. An all-optical point-to-point connection between two nodes is called a *lightpath*. Each lightpath is characterized by its route and by the wavelength with which it is multiplexed. Two lightpaths may use the same wavelength, provided they do not share any common fiber.

Given the physical topology of an optical network and a set of lightpaths defining a logical topology in this network, the problem of *Routing and Wavelength Assignment* (RWA) in WDM optical networks consists in routing the set of lightpaths and assigning a wavelength to each of them, such that lightpaths whose routes share a common fiber are assigned to different wavelengths. Variants of RWA are characterized by different optimization criteria and traffic patterns, see e.g. [11,57]. We consider RWA variants in which the lightpath requests are known beforehand and no wavelength conversion is available, i.e. a lightpath must be assigned to the same wavelength on all fibers along its route. Let $G = (V, A)$ be a directed graph representing the physical network topology, where V is the set of the nodes and A represents the fiber connections between the nodes. Let also R be the set of lightpaths requests, each one defined by a source and a destination node in V. There can be more than one lightpath request between any pair of nodes, since the traffic between a pair of nodes can be larger than that supported by a single lightpath. We denote by λ the number of available wavelengths. In the min-RWA problem variant [14], the number of available wavelengths is unbounded and the objective is to minimize the number of wavelengths used to establish all lightpath requests in R. This paper focus on the max-RWA variant [10], where $\lambda < |R|$. Therefore, it may not be possible to accept and route all requests in R. The objective is to maximize the number of requests that may be accepted. Both problems have been proved to be NP-hard in [14] and [10], respectively.

Exact algorithms based on integer programming formulations have been proposed in the literature. However, as the worst case complexity of these algorithms grows exponentially with the size of the network, they can only solve small instances to optimality. State-of-the-art exact algorithms can only provide upper bounds to the optimal value for the largest instances in the literature. To the best of our knowledge, the only existing heuristics for max-RWA are the greedy algorithms in [33,38], the graph partitioning approach in [5], and the tabu search heuristic in [12].

In this paper, we propose heuristics for efficiently solving large instances in the literature. Related work is reviewed in Sect. 2. Greedy constructive heuristics are proposed in Sect. 3. A biased random-key genetic algorithm (BRKGA), based on the best of the greedy heuristics, is proposed in Sect. 4. Computational experiments are reported in Sect. 5, where it is shown that the proposed heuristics outperform the previous heuristics in the literature. Finally, concluding remarks are drawn in the last section.

2 Related work

2.1 Exact formulations and algorithms for max-RWA

Most of the literature on algorithms for solving max-RWA reports on integer programming formulations and exact approaches [26,30,33,33,34,41,48]. A review on formulations for max-RWA can be found in [27,29,42].

Krishnaswamy and Sivarajan [33] developed an arc based compact formulation for max-RWA. This formulation allows cycles, but they argue that these cycles have no impact in the value of the objective function. A cycle free compact formulation based on the previous one was proposed in [26]. This formulation does not improve the upper bounds of the former, but it can be solved more efficiently by MIP solvers. Martins [42] improved the compact formulations of [26,33] and found optimal solutions for instances with up to 18 nodes.

Lee et al. [34] proposed a formulation based on the weighted independent set problem with additional cardinality constraints [21], whose linear relaxation can be solved by a column generation algorithm. Jaumard et al. [26,30] proposed an improved formulation that uses only maximal independent sets and found better upper bounds than those of [34] for instances with up to 27 nodes.

Ramaswami and Sivarajan [48] proposed a maximum independent set formulation with an exponential number of variables. They proved that the upper bound provided by its linear relaxation is never smaller than that of the best compact integer programming formulations found in the literature at the time of its publication. Jaumard et al. [30] implemented a column generation algorithm based on this formulation and solved instances with up to 27 nodes to optimality.

Martins et al. [41,42] proposed a new, improved formulation based on those in [30,34,48]. Their approach obtained the first upper bounds for the three instances based on the 71-node ATT2 network, that could not be computed by previous works in the literature due to memory overflow. They found the best upper bounds for the largest instances in the literature of max-RWA with up to 90 nodes. However, not even feasible solutions have been reported for these instances.

2.2 Heuristics for max-RWA

To the best of our knowledge, there are few heuristics for max-RWA in the literature. Krishnaswamy and Sivarajan [33] proposed two rounding heuristics based on the linear relaxation of their integer programming formulations. Computational experiments carried out on two networks with 14 and 20 nodes showed that the average relative optimality gap for the best of the two heuristics was of 6.0 and 7.2 %, respectively.

Manohar, Manjunath, and Shevgaonkar [38] developed the `Greedy-EDP-RWA` heuristic. At each iteration, a subset of lightpaths is selected and routed with edge disjoint paths by the `BGAforEDP` heuristic for the *maximum edge disjoint path* (EDP) problem [32]. Then, all lightpaths in this subset are assigned the same wavelength, and the procedure is repeated with the remaining lightpaths. This heuristic was proposed for the min-RWA variant of the problem, but the authors argue that it can also be used for max-RWA, by running BGAforEDP for λ iterations. The authors reported that their algorithm was faster than and found solutions as good as the linear programming based algorithms for min-RWA at the time of its publication. No computational experiments on the performance of this heuristic for max-RWA have been reported by the authors.

A tabu search heuristic was proposed by Dzongang et al. [12]. First, this heuristic builds a set P_r of pre-computed paths between the endnodes of each request $r \in R$. Each neighbor of the current solution S is generated by using a path $p \in P_r$ and a wavelength $\omega \in \{1, \ldots, \lambda\}$ to route a request r not accepted in the solution S and then removing from S all requests assigned to wavelength ω that share a link with p. Computational experiments on the same instances used in [33] have shown that tabu search found much better solutions than the rounding heuristics of [33], with average relative optimality gaps of 1.41 and 1.53 % for the instances with 14 and 20 nodes, respectively.

Belgacem and Puech [5] proposed a decomposition heuristic for max-RWA, in which the original instance is partitioned into smaller instances which are exactly solved by integer programming. The local solutions are combined into a feasible solution. This proposal was validated by an application to the European backbone network EBN57 with 57 nodes and to randomly generated planar networks with up to 500 nodes.

2.3 Heuristics for variants of max-RWA with wavelength conversion

Marković et al. [39] proposed a bee colony optimization heuristic for the variant of max-RWA in which the wavelength conversion is available at some nodes. Computational experiments showed that the heuristic was able to produce optimal or near-optimal solutions only for two small networks with 8 and 18 nodes.

Qin et al. [47] proposed a new optimization objective, which consists in determining the maximum number of connections with the least number of wavelength converters. The problem was tackled by a genetic algorithm.

Jaumard et al. [28] proposed a new two-phase heuristic for the variant in which wavelength conversion is possible in every node. Wavelength assignment is reformulated as a generalized partition coloring problem, extending the partition coloring formulation proposed by Noronha and Ribeiro [46]. This problem was solved by a tabu search heuristic. Computational experiments on instances with up to 27 nodes have shown that wavelength conversion is of little help to increase the number of accepted lightpaths, except for some very particular traffic patterns [28].

2.4 Heuristics for min-RWA

Contrarily to the max-RWA variant considered in this paper, most of the research on algorithms for solving the min-RWA variant is focused into heuristics. Some approaches decompose the problem into two subproblems (routing and wavelength assignment) [3,24,25,36,46], while others tackle the two subproblems simultaneously [38,40–43,45,54].

Skorin-Kapov [54] proposed the current state-of-the-art greedy constructive heuristics for min-RWA. Each wavelength is associated with a different copy of G. Lightpaths that are routed along disjoint arcs on the same copy of G are assigned to the same wavelength. Copies of G are associated with the bins and lightpaths with the items of an instance of the *bin packing problem* [31]. In this context, min-RWA can be reformulated as the problem of packing all the lightpath requests (items) in a minimum number of wavelength (bins). Four min-RWA heuristics based on classical bin packing heuristics were developed: (i) FF-RWA,

```
procedure BFD-RWA(G,R,d,π)
1   Set S ← ∅, Ω ← ∅, and k ← 0;
2   for i = 1,...,|R| do
3       if there is a path for routing lightpath π(i) with less than d arcs in any of the graphs G_1,...,G_k
4       then
5           Let r ∈ {1,...,k} such that lightpath π(i) can be routed in graph G_r with the
            least number of arcs;
6       else
7           k ← k+1;
8           G_k ← G;
9           r ← k;
10      end-if;
11      Let P_π(i) be the shortest path between the endnodes of lightpath π(i) in G_r;
12      S ← S ∪ {(p_π(i),r)};
13      Remove all arcs in path P_π(i) from G_r;
14  end-for;
15  return S,k;
end BFD-RWA.
```

Fig. 1 Pseudo-code of the BFD-RWA heuristic for min-RWA

based on the *first fit* heuristic, (ii) BF-RWA, based on the *best fit* heuristic, (iii) FFD-RWA, based on the *first fit decreasing* heuristic, and (iv) BFD-RWA, based on the *best fit decreasing* heuristic.

Computational results have shown that the best results have been obtained by BFD-RWA, whose pseudo-code is presented in Fig. 1. The inputs are the graph G, the set R of lightpath requests, the value d of the maximum number of arcs allowed in each route, and the vector $\pi = (\pi(1), \dots, \pi(|R|))$ describing the order in which the lightpaths will be considered. Let *min-length*(i) be the number of hops in the path in G with the smallest number of arcs between the endnodes of lightpath $i \in R$. The lightpaths will be considered in a non-increasing order of their *min-length* values, i.e., $\pi(k) = \text{argmax}\{\textit{min-length}(i) : i \in R \setminus \{\pi(1), \dots, \pi(k-1)\}\}$, for any $k = 1, \dots, |R|$. Ties between two lightpaths with the same min-length value are broken arbitrarily. The idea behind this ordering is that long lightpaths are harder to be routed and therefore should be routed first. The output is a set S of pairs in which the first element is the path used to route lightpath i and the second is the wavelength with which it is multiplexed.

Each solution is formed by a set of pairs, each of them containing the route and the wavelength assigned to one lightpath. The solution S and the number k of lightpaths used are initialized in line 1. The lightpaths are routed one at a time (and assigned to a wavelength) in lines 2 to 14. In line 3, the algorithm determines whether lightpath $\pi(i)$ can be routed using any of the k wavelengths already used. If this is the case, then in line 5 the algorithm determines wavelength r as that in which lightpath $\pi(i)$ can be routed with the least number of arcs. Otherwise, the number of wavelengths used is increased by one in line 7, a new copy of graph G is created in line 8, and the new wavelength $r = k$ is selected to be assigned to lightpath $\pi(i)$ in line 9. Line 11 computes the shortest path $P_{\pi(i)}$ between the endnodes of lightpath $\pi(i)$ in G_r. In line 12, the pair $(P_{\pi(i)}, r)$ is added to the solution under construction and all arcs in route $P_{\pi(i)}$ used by lightpath $\pi(i)$ are removed from G_r in line 13. A feasible solution S and the number k of wavelengths used are returned in line 15.

Noronha et al. [44] developed efficient algorithms and data structures for the implementation of heuristics FF-RWA, BF-RWA, FFD-RWA, and BFD-RWA. The longest running times of their implementation of BFD-RWA took less than 3 s, while the times reported for the heuristic in [54] took up to 8 min on the same instances and the same Pentium IV 2.8 GHz processor. BFD-RWA was successfully used in the iterated local search heuristic of Martins at al. [43] and in the biased random-key genetic algorithm of Noronha et al. [45] for the problem min-RWA. Computational experiments have shown that these are to date the best heuristics for min-RWA. These results motivated the development of constructive heuristics based on BFD-RWA and the development of a biased random-key genetic algorithm for the max-RWA problem variant.

3 Greedy constructive heuristics for max-RWA

State-of-the-art exact algorithms and heuristics for max-RWA perform well only for small instances with 27 nodes. In Sect. 3.1, we present three greedy heuristics for max-RWA based on bin packing that are extensions of the best heuristic (BFD-RWA) for min-RWA. Next, we propose in Sect. 3.2 three new greedy heuristics for max-RWA that are based on multiprocessor scheduling. Later, we will show that the heuristics contribute effectively in the solution of the largest instances in the literature of max-RWA with up to 90 nodes.

3.1 Constructive heuristics based on bin packing

We recall that for min-RWA one may use as many wavelengths as necessary, while for max-RWA there are at most λ available wavelengths for routing. The three heuristics (BFR-RWA, BFI-RWA, and BFD-RWA) proposed in this section are derived from BFD-RWA and differ by the order in which the lightpaths are considered:

– BFR: requests are randomly selected from R;
– BFI: requests are sorted in non-decreasing order of their min-length values; and
– BFD: as for BFD-RWA, requests are sorted in non-increasing order of their min-length values, with ties arbitrarily broken.

A general framework BF* for the pseudo-code of the family of heuristics BFR, BFI, and BFD is given in Fig. 2. The inputs are the graph G, the set R of lightpath requests, the maximum number d of arcs in each route, a permutation $\pi(1), \ldots, \pi(|R|)$ describing the order in which the requests are considered, and the maximum number λ of wavelengths available. As for min-RWA, each solution is formed by a set of pairs, each of them containing the route and the wavelength assigned to one lightpath. The solution S and the number k of lightpaths used are initialized in line 1. The lightpaths are routed one at a time (and assigned to a wavelength) in lines 2 to 14. In line 3, sets r to zero as a flag in case lightpath $\pi(i)$ can not be the routed with the λ available wavelengths. The algorithm determines in line 4 whether lightpath $\pi(i)$ can be routed using any of the k wavelengths already used. If this is the case, then in line 6 the algorithm determines wavelength r as that in which lightpath $\pi(i)$ can be routed with the least number of arcs. Otherwise, and if not all wavelengths have already been used, the number of wavelengths used is increased by one in line 8, a new copy of graph G is created in line 9, and the new wavelength $r = k$ is selected to be assigned to lightpath $\pi(i)$ in line 10. If lightpath $\pi(i)$ can be routed, then line 14 computes the shortest path $P_{\pi(i)}$ between the endnodes of lightpath $\pi(i)$ in G_r. In line 15, the pair $(P_{\pi(i)}, r)$ is added to the solution under construction and all arcs in route $P_{\pi(i)}$ used by lightpath $\pi(i)$ are removed from G_r in line 16. A feasible solution S and the number k of wavelengths used are returned in line 19.

```
procedure BF*(G,R,d,π,λ)
    Set S ← ∅ and k ← 0;
    for i = 1,...,|R| do
        r ← 0;
        if there is a path for routing lightpath π(i) with less than d arcs in any of the graphs G_1,...,G_k
        then
            Let r ∈ {1,...,k} such that lightpath π(i) can be routed in graph G_r with the
            least number of arcs;
        else if k < λ then
                k ← k+1;
                G_k ← G;
                r ← k;
            end-if;
        end-if;
        if r > 0 then
            Let P_π(i) be the shortest path between the endnodes of lightpath π(i) in G_r;
            S ← S ∪ {(P_π(i), r)};
            Remove all arcs in path P_π(i) from G_r;
        end-if
    end-for;
    return S,k;
end BF*.
```

Fig. 2 Pseudo-code of the family BF* of constructive heuristics for max-RWA based on bin packing

3.2 Constructive heuristics based on multi-processor scheduling

In this section, we propose three new heuristics for max-RWA based on a classic heuristic for the *Multi-Processor Scheduling Problem* (MPSP) [20]. Given a set P of processors and a set T of tasks, where each task $t \in T$ is associated with a running time b_t, the heuristic consists in assigning a processor $p_t \in P$ to each task $t \in T$, in order to minimize the *maximum completion time* of a processor. If $T_p \subseteq T$ denotes the subset of tasks assigned to each processor $p \in P$ in a given solution, the maximum completion time of this solution is given by $\max_{p \in P}\{\sum_{t \in T_p} b_t\}$.

The *Longest Processing Time* heuristic [23] builds a feasible solution for MPSP as follows. First, it sorts the tasks in non-increasing order of their completing time. Following this order, each task is scheduled to one of the available processors in which the maximum completing time increases the least.

The three heuristics for max-RWA described below are inspired by this algorithm. Each wavelength is associated with a different copy of G. Lightpaths that are routed along disjoint arcs on the same copy of G are assigned the same wavelength. Each copy of G is associated with one processor and each lightpaths with one of the tasks of an instance of MPSP. The completion time of each task is defined as the min-length of the corresponding lightpath. Therefore, max-RWA can be reformulated as the problem of scheduling the tasks (lightpath requests) in the set of processors (wavelengths).

The pseudo-code of the new family *PT of heuristics is presented in Fig. 3. As for BFR, BFI and BFD, the inputs are the graph G, the maximum number d of arcs in each route, the set R of lightpath requests, a permutation $\pi(1), \ldots, \pi(|R|)$ describing the order in which the lightpaths are considered, and the maximum number λ of wavelengths available. Once again, the three heuristics (RPT, SPT, and LPT) differ by the order in which the lightpaths are considered:

– RPT: requests are randomly selected from R;

procedure *PT(G, R, d, π, λ)
Set $S \leftarrow \emptyset$ and $k \leftarrow 0$;
for $k = 1, \dots, \lambda$ **do**
  $G_k \leftarrow G$;
end-for;
for $i = 1, \dots, |R|$ **do**
  if lightpath $\pi(i)$ can be routed with less than d arcs in any of the graphs $G_1, \dots, G_\lambda$
  then
    $k \leftarrow k + 1$;
    Let $r \in \{1, \dots, \lambda\}$ such that lightpath $\pi(i)$ can be routed in graph G_r with the least number of arcs;
    Let $P_{\pi(i)}$ be the shortest path between the endnodes of lightpath $\pi(i)$ in G_r;
    $S \leftarrow S \cup \{(P_{\pi(i)}, r)\}$;
    Remove all arcs in path $P_{\pi(i)}$ from G_r;
  end-if
end-for;
return S, k;
end *PT.

Fig. 3 Pseudo-code of the family *PT of constructive heuristics for max-RWA based on multi-processor scheduling

- SPT: requests are sorted in non-decreasing order of their min-length values; and
- LPT: as for the Longest Processing Time heuristic, requests are sorted in non-increasing order of their min-length values, with ties arbitrarily broken.

Algorithms *PT in Fig. 3 are very similar to those named as BF* in Fig. 2. Basically, the only difference consists that in the family *PT all processors are available (or, in other words, all copies of the original graph are built beforehand) since the beginning of the algorithm. As the numerical results will show, the underlying strategy of *PT is better, since creating all copies of the graphs beforehand makes it possible to assign the best route (i.e., that with the least number of edges) among all copies of the graph to each lightpath. Consequently, the edges blocked to be used for each lightpath will be the least possible and will leave more choices for the lightpaths that will be considered next.

Heuristics BFR, BFI, BFD, RPT, SPT, LPT, and GREEDY-EDP-RWA [38] will be evaluated and compared in Sect. 5.

4 Biased random-key genetic algorithm

Genetic algorithms with random keys, or random-key genetic algorithms (RKGA), were first introduced by Bean [4] for combinatorial optimization problems for which solutions can be represented as a permutation vector. Solutions are represented as vectors of randomly generated real numbers called keys. A deterministic algorithm, called a decoder, takes as input a solution vector and associates with it a feasible solution of the combinatorial optimization problem, for which an objective value or fitness can be computed. Two parents are selected at random from the entire population to implement the crossover operation in the implementation of a RKGA. Parents are allowed to be selected for mating more than once in a given generation.

A biased random-key genetic algorithm (BRKGA) differs from a RKGA in the way parents are selected for crossover, see Gonçalves and Resende [18] for a review. In a BRKGA, each element is generated combining one element selected at random from the elite solutions in the current population, while the other is a non-elite solution. We say the selection is biased since one parent is always an elite individual and because this elite solution has a higher probability of passing its genes to the offsprings, i.e. to the new generation. A BRKGA provides a better implementation of the essence of Darwin's principle of "survival of the fittest", since an elite solution has a higher probability of being selected for mating and the offsprings have a higher probability of inheriting the genes of the elite parent.

The development and application of a BRKGA to max-RWA was motivated by successful applications to many network optimization problems, such as other variants of routing and wavelength assignment [19,44–46], routing in OSPF networks [7,8,13,49], road congestion [9], as well as to other combinatorial optimization problems, such as job shop scheduling [15,35], assembly line balancing [16], lateness minimization on parallel batch processing machines [37,56], and cell formation in manufacturing [17], among others.

The biased random-key genetic algorithm for max-RWA evolves a population of chromosomes that consists of vectors of real numbers (called keys). Each solution is represented by an $|R|$-vector, in which each component is a real number in the range [0, 1] associated with a lightpath request in R. Each solution represented by a chromosome is decoded by a decoding heuristic that receives the vector of keys and builds a feasible solution for max-RWA.

Any of the heuristics BF* or *PT presented in Sect. 3 can be used as a decoding heuristic for BRKGA. Since the computational experiments that will be presented in the next section have shown that SPT obtains the best results among the seven tested greedy constructive heuristics, the description of BRKGA that follows will consider SPT as the decoder. The decoding consists of two steps. First, the lightpaths are sorted in non-decreasing order of the sum of their min-length and key values. Therefore, the relative order between lightpaths with the same min-length value is defined by their keys. The resulting order is used as the vector π in SPT (see the pseudo-code in Fig. 3). The number of wavelengths found by SPT using this order is used as the fitness of the chromosome. The algorithm stops when a maximum elapsed time is reached or when a solution as good as a given target is found.

We use the parametrized uniform crossover scheme proposed in [55] to combine two parent solutions and produce an offspring. In this scheme, the offspring inherits each of its keys from the best fit of the two parents with probability $\rho > 0.5$ and from the least fit parent with probability $1-\rho$. This genetic algorithm does not make use of the standard mutation operator, where parts of the chromosomes are changed with small probability. Instead, the concept of mutants is used: a fixed number of mutant solutions are introduced in the population in each generation, randomly generated in the same way as in the initial population. Mutants play the same role of the mutation operator in traditional genetic algorithms, diversifying the search and helping the procedure to escape from locally optimal solutions.

The keys associated to each lightpath request are randomly generated in the initial population. At each generation, the population is partitioned into two sets: TOP and $REST$. Consequently, the size of the population is $|TOP| + |REST|$. Subset TOP contains the best solutions in the population. Subset $REST$ is formed by two disjoint subsets: MID and BOT, with subset BOT being formed by the worst elements on the current population. As illustrated in Fig. 4, the chromosomes in TOP are simply copied to the population of the next generation. The elements in BOT are replaced by newly created mutants that are placed in the new set BOT. The remaining elements of the new population are obtained by crossover with one parent randomly chosen from TOP and the other from $REST$. This distinguishes a biased random-key GA from the random-key genetic algorithm of Bean [4] (where both parents are selected at random from the entire population). Since a parent solution can be chosen for crossover more than once in a given generation, elite solutions have a higher probability of passing their random keys to the next generation. In this way, $|MID| = |REST| - |BOT|$ offspring solutions are created.

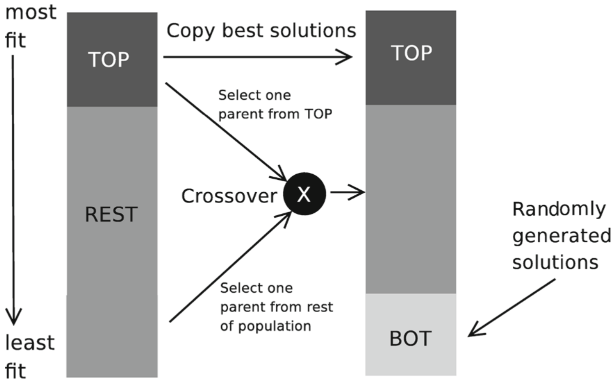

Fig. 4 Population evolution between consecutive generations of a BRKGA

5 Computational experiments

The heuristics Greedy-EDP-RWA [38], BFD, BFR, BFI, LPT, RPT, SPT, and BRKGA were implemented in C++ and compiled with GNU C++ version 4.6.3. The inheritance probability ρ of the crossover operator in BRKGA was set to 0.7, as used and recommended in [6,15,19,45,52,58]. The population size was set to $|TOP| + |MID| + |BOT| = |V|$, with the sizes of sets TOP, $REST$, and BOT set to $0.25 \times |V|$, $0.7 \times |V|$, and $0.05 \times |V|$, respectively, once again as suggested and used in [19,45] for the problem of routing and wavelength assignment. The experiments were performed on a 3.40 GHz i7-4770 Intel Core CPU with 16 GB of RAM memory.

Four sets of instances have been used in the experiments. The physical topologies are connected and each link corresponds to a pair of bidirectional fibers. The logical topology is asymmetric, i.e. there might be a lightpath request from a node i to a node j, while not from j to i. The first three sets were proposed in [41] and have 24 instances each. They are based in the same 24 networks with up to 90 nodes, but differ from each other by the number of wavelengths available. Set A is formed by instances in which there are ten wavelengths available. Instances in sets B and C have 20 and 30 wavelengths available, respectively. These are the largest and most difficult instances in the literature of max-RWA. No feasible solutions are available in the literature, only the upper bounds obtained by the column generation algorithm of [42]. The main characteristics of the networks used in sets A, B, and C are presented in Table 1, which gives respectively, the network name, the number of lightpath requests, the number of nodes, and the number of arcs of each network.

The fourth set of instances, referred here as D, was used by Dzongang et al. [12] to evaluate the performance of their tabu search heuristic, and to show that the latter outperforms the two rounding heuristics of [33]. It has 30 instances based on the NSF and EONNET networks. These networks have the same physical topologies of the networks NSF1 and EON of Table 1, respectively, but have different sets of lightpath requests and different values for λ. NSF has 14 nodes, 21 links, and 268 lightpath requests, while EONNET has 20 nodes, 39 links, and 374 lightpath requests. Fifteen instances are based on NFS and have from 10 to 24 wavelengths available, while the other fifteen instances are based on EONNET and also have from 10 to 24 wavelengths available.

In the first experiment, we evaluate and compare the quality of the solutions provided by the heuristics Greedy-EDP-RWA, BFD, BFR, BFI, LPT, RPT, and SPT for the instances in

Table 1 Main characteristics of the networks in sets A, B, and C

Name	Lightpaths	Nodes	Links
ATT	359	90	137
ATT2	2918	71	175
BRASIL	1370	27	70
COST266	6543	37	57
DFN-BWIN	4840	10	45
DFN-GWIN	3771	11	47
EON	373	20	39
FINLAND	930	31	51
FRANCE	15,398	25	45
GIUL	14,732	39	86
JANOS-US	3262	26	42
NOBEL-EU	1898	28	41
NOBEL-GERMANY	660	17	26
NOBEL-US	478	14	21
NORWAY	5348	27	51
NSF1	284	14	21
NSF12	551	14	21
NSF21	284	14	22
NSF212	551	14	22
NSF23	285	14	22
NSF248	547	14	22
NSF3	285	14	21
NSF48	547	14	21
SUN	952	27	51

sets A, B, and C. The vectors π describing the order in which the lightpaths are considered by heuristics Greedy-EDP-RWA, BFR, and RPT have been randomly generated. Ties between two lightpaths with the same min-length value were broken randomly for heuristics BFD, BFI, LPT, and SPT. Each heuristic was run 10 times for each instance, with different seeds for the random number generator [53]. The average optimality gap (UB–LB/UB between the value LB of the solution provided by the heuristic and the upper bound UB presented in [42] is calculated for each instance.

The average gap over all instances in each set is displayed in Fig. 5 for each constructive heuristic. We first observe that the relative behavior of the seven heuristics is absolutely the same for the three test sets. We notice that BFD and LPT led to the largest average gaps among the seven tested heuristics. This is due to the fact that in these algorithms the lightpaths requests are sorted in non-increasing order of their min-length values, that is, the hardest lightpath to route is the first inserted into the solution. Although this greedy criterion often leads to good results for other problems, in situations such as max-RWA it might be better not to accept long lightpaths (i.e., those with large min-length values) in order to save space for shorter lightpaths (i.e., with smaller min-length values). The best results have been obtained by heuristics BFI and SPT for all test sets, with both of them considering the lightpath requests in a non-decreasing order of their min-length values. The

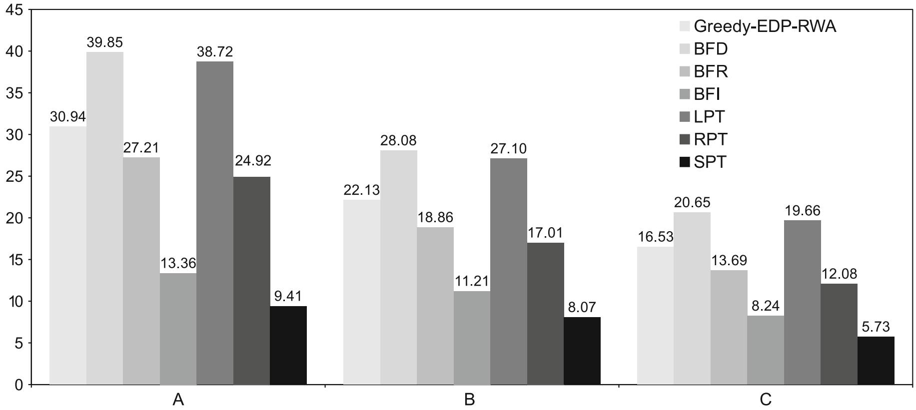

Fig. 5 Average relative optimality gaps for heuristics Greedy-EDP-RWA[38], BFD, BFR, BFI, LPT, RPT, SPT for instance sets A, B, and C

average optimality gaps observed for BFI and SPT were, respectively, 13.36 and 9.41 % for set A, 11.21 and 8.07 % for set B, and 8.24 and 5.73 % for set C. Heuristic SPT proposed in this work outperformed the BFI heuristic based on the work of [54]. In addition, the average optimality gaps provided by SPT were less than half of those observed with the Greedy-EDP-RWA [38] heuristic, which amounted to 30.94, 22.13, and 16.53 % for sets A, B, and C, respectively. These results indicate that SPT is the most promising heuristic to be used in advanced metaheuristics for max-RWA. Therefore, SPT will be used as the decoder for our implementation of the BRKGA heuristic for max-RWA.

In the second experiment, we investigate if BRKGA with SPT as its decoder efficiently identifies the relationships between keys and good solutions and converges to near-optimal solutions. For this end, we compare its performance with that of a multi-start (MS) procedure that uses the same decoding heuristic as BRKGA. Each iteration of the multi-start procedure applies this same decoding heuristic starting from a randomly generated vector of random-keys. Therefore, nothing is learned from one iteration of MS to the next.

Each of the heuristics BRKGA and MS was given 10 min of computation time and stopped thereafter. Ten runs of each heuristic have been performed for each instance, with different seeds for the random number generator [53]. The results for instance sets A, B, and C are displayed in Tables 2, 3 and 4. The first two columns provide the network name and the upper bound (*UB*) already reported in [41]. The minimum (*min*), average (*avg*), and maximum (*avg*) solution values obtained by MS are displayed in the next three columns. The average optimality gap, and the coefficient of variation ($CV = \sigma/avg$) are shown in the following two columns, where σ is the standard deviation of the sample. The same statistics are displayed for BRKGA in the subsequent columns. The best average optimality gaps for each instance are displayed in boldface.

Table 2 shows that BRKGA found better solutions than MS for all instances in set A. The average optimality gap observed for BRKGA was only 3.54 %, while that for MS was 5.45 %. Besides, the maximum gap for MS was 10.38 %, while that for BRKGA was only 7.30 %. Similar results were observed for sets B and C. Table 3 shows that MS never obtained better average solution values than BRKGA, which found strictly better solutions than MS for all, but three instances in set B. The average optimality gap obtained with BRKGA for this set was only 3.99 %, while that resulting from MS was 5.49 %. The maximum gap for

Table 2 BRKGA versus multi-start for instances of test set A

Instance	*UB*	MS					BRKGA				
		min	*avg*	*max*	Gap (%)	CV (%)	*min*	*avg*	*max*	Gap (%)	CV (%)
ATT	253.00	234.00	234.60	235.00	7.27	0.11	236.00	239.80	244.00	**5.22**	0.31
ATT2	895.00	856.00	856.60	858.00	4.29	0.21	886.00	887.40	889.00	**0.85**	0.29
BRASIL	721.50	662.00	664.30	666.00	7.93	0.23	677.00	680.40	686.00	**5.70**	0.60
COST266	788.50	735.00	737.10	740.00	6.52	0.33	750.00	754.40	762.00	**4.32**	0.81
DFN-BWIN	884.00	809.00	811.10	814.00	8.25	0.35	837.00	844.10	853.00	**4.51**	1.08
DFN-GWIN	874.00	802.00	803.10	805.00	8.11	0.22	827.00	834.70	843.00	**4.50**	0.99
EON	285.00	279.00	279.30	280.00	2.00	0.11	280.00	281.20	282.00	**1.33**	0.13
FINLAND	444.77	408.00	408.70	411.00	8.11	0.21	415.00	420.40	424.00	**5.48**	0.60
FRANCE	610.00	545.00	546.70	550.00	10.38	0.31	557.00	565.50	573.00	**7.30**	1.28
GIUL	1561.00	1477.00	1480.20	1486.00	5.18	0.60	1519.00	1524.00	1527.00	**2.37**	0.62
JANOS-US	600.00	584.00	585.20	587.00	2.47	0.20	587.00	591.30	594.00	**1.45**	0.43
NOBEL-EU	346.00	330.00	330.20	332.00	4.57	0.14	334.00	336.20	338.00	**2.83**	0.24
NOBEL-GERMANY	228.00	225.00	225.60	226.00	1.05	0.11	227.00	227.70	228.00	**0.13**	0.10
NOBEL-US	190.00	187.00	187.90	188.00	1.11	0.07	189.00	189.80	190.00	**0.11**	0.09
NORWAY	834.00	795.00	795.80	797.00	4.58	0.17	806.00	809.70	814.00	**2.91**	0.52
NSF1	197.00	186.00	186.20	187.00	5.48	0.09	187.00	187.70	189.00	**4.72**	0.14
NSF12	264.00	246.00	247.10	248.00	6.40	0.16	252.00	252.80	254.00	**4.24**	0.17
NSF21	205.00	195.00	195.60	196.00	4.59	0.11	196.00	196.70	198.00	**4.05**	0.14
NSF212	280.33	261.00	261.40	263.00	6.75	0.15	264.00	267.00	269.00	**4.76**	0.30
NSF23	206.00	194.00	194.50	195.00	5.58	0.12	196.00	197.40	199.00	**4.17**	0.18
NSF248	266.33	251.00	251.70	252.00	5.49	0.11	254.00	256.50	258.00	**3.69**	0.31
NSF3	195.50	183.00	183.10	184.00	6.34	0.07	184.00	184.90	186.00	**5.42**	0.01
NSF48	254.00	239.00	239.90	241.00	5.55	0.12	243.00	244.20	246.00	**3.86**	0.26
SUN	264.00	256.00	256.70	257.00	2.77	0.11	258.00	261.10	262.00	**1.10**	0.29
Average					5.45					3.54	

The smallest optimality gap for each instance is displayed in boldface

Table 3 BRKGA versus multi-start for instances of test set B

Instance	*UB*	MS					BRKGA				
		min	*avg*	*max*	Gap (%)	CV (%)	*min*	*avg*	*max*	Gap (%)	CV (%)
ATT	359.00	324.00	324.90	326.00	9.50	0.08	331.00	333.80	336.00	**7.02**	0.02
ATT2	1298.00	1210.00	1211.60	1213.00	6.66	0.11	1238.00	1240.80	1245.00	**4.41**	0.29
BRASIL	1080.67	1017.00	1018.40	1023.00	5.76	0.32	1026.00	1030.40	1033.00	**4.65**	0.49
COST266	1325.00	1197.00	1199.50	1205.00	9.47	0.32	1232.00	1238.00	1243.00	**6.57**	0.49
DFN-BWIN	1733.00	1615.00	1618.20	1623.00	6.62	0.33	1657.00	1665.70	1671.00	**3.88**	0.64
DFN-GWIN	1519.00	1464.00	1466.00	1470.00	3.49	0.23	1492.00	1498.60	1502.00	**1.34**	0.42
EON	369.00	367.00	367.00	367.00	0.54	0.00	368.00	368.40	369.00	**0.16**	0.07
FINLAND	642.00	607.00	607.40	608.00	5.39	0.07	609.00	613.10	617.00	**4.50**	0.38
FRANCE	1181.00	1069.00	1072.40	1078.00	9.20	0.34	1097.00	1103.20	1111.00	**6.59**	0.61
GIUL	2519.00	2365.00	2367.40	2370.00	6.02	0.23	2421.00	2430.10	2440.00	**3.53**	0.86
JANOS-US	981.00	913.00	914.70	917.00	6.76	0.16	927.00	933.90	938.00	**4.80**	0.45
NOBEL-EU	596.00	561.00	561.70	564.00	5.76	0.13	567.00	572.10	576.00	**4.01**	0.33
NOBEL-GERMANY	384.00	363.00	364.60	366.00	5.05	0.16	367.00	369.90	371.00	**3.67**	0.17
NOBEL-US	326.50	311.00	311.80	313.00	4.50	0.09	314.00	314.90	317.00	**3.55**	0.13
NORWAY	1377.00	1262.00	1263.80	1266.00	8.22	0.15	1289.00	1296.10	1303.00	**5.88**	0.58
NSF1	278.00	272.00	272.00	272.00	**2.16**	0.00	272.00	272.00	272.00	**2.16**	0.00
NSF12	408.00	381.00	382.30	383.00	6.30	0.09	386.00	387.40	389.00	**5.05**	0.15
NSF21	282.00	282.00	282.00	282.00	**0.00**	0.00	282.00	282.00	282.00	**0.00**	0.00
NSF212	427.00	403.00	403.00	403.00	5.62	0.00	407.00	408.60	411.00	**4.31**	0.18
NSF23	284.00	280.00	280.00	280.00	**1.41**	0.00	280.00	280.00	280.00	**1.41**	0.00
NSF248	413.00	383.00	384.20	385.00	6.97	0.09	389.00	392.80	395.00	**4.89**	0.23
NSF3	277.00	273.00	273.00	273.00	1.44	0.00	273.00	273.30	274.00	**1.34**	0.06
NSF48	389.00	368.00	368.78	370.00	5.20	0.09	372.00	374.33	378.00	**3.77**	0.24
SUN	502.00	452.00	453.20	455.00	9.72	0.14	459.00	460.70	462.00	**8.23**	0.14
Average					5.49					3.99	

The smallest optimality gap for each instance is displayed in boldface

Table 4 BRKGA versus multi-start for instances of test set C

Instance	*UB*	MS					BRKGA				
		min	*avg*	*max*	gap (%)	CV (%)	*min*	*avg*	*max*	gap (%)	CV (%)
ATT	359.00	359.00	359.00	359.00	**0.00**	0.00	359.00	359.00	359.00	**0.00**	0.00
ATT2	1648.00	1517.00	1518.60	1520.00	7.85	0.12	1558.00	1581.40	1603.00	**4.04**	1.78
BRASIL	1241.00	1233.00	1234.10	1235.00	0.56	0.06	1236.00	1237.00	1238.00	**0.32**	0.08
COST266	1728.94	1567.00	1567.60	1569.00	9.33	0.07	1592.00	1600.60	1609.00	**7.42**	0.63
DFN-BWIN	2555.50	2407.00	2409.20	2413.00	5.72	0.17	2442.00	2460.20	2478.00	**3.73**	1.27
DFN-GWIN	1945.00	1943.00	1944.40	1945.00	0.03	0.07	1945.00	1945.00	1945.00	**0.00**	0.00
EON	373.00	373.00	373.00	373.00	**0.00**	0.00	373.00	373.00	373.00	**0.00**	0.00
FINLAND	774.00	748.00	749.30	752.00	3.19	0.11	752.00	754.50	760.00	**2.52**	0.24
FRANCE	1730.00	1570.00	1573.40	1579.00	9.05	0.29	1598.00	1607.00	1615.00	**7.11**	0.57
GIUL	3307.50	3058.00	3065.00	3078.00	7.33	0.59	3130.00	3140.70	3160.00	**5.04**	0.91
JANOS-US	1239.00	1172.00	1172.30	1173.00	5.38	0.05	1187.00	1189.50	1193.00	**4.00**	0.23
NOBEL-EU	822.00	757.00	757.70	760.00	7.82	0.10	769.00	772.00	776.00	**6.08**	0.16
NOBEL-GERMANY	489.00	462.00	462.70	465.00	5.38	0.10	465.00	467.10	469.00	**4.48**	0.16
NOBEL-US	432.50	403.00	404.00	405.00	6.59	0.05	407.00	409.50	412.00	**5.32**	0.14
NORWAY	1782.00	1635.00	1636.90	1639.00	8.14	0.14	1663.00	1675.80	1683.00	**5.96**	0.58
NSF1	284.00	284.00	284.00	284.00	**0.00**	0.00	284.00	284.00	284.00	**0.00**	0.00
NSF12	499.00	476.00	476.30	477.00	4.55	0.05	477.00	480.70	482.00	**3.67**	0.14
NSF21	284.00	284.00	284.00	284.00	**0.00**	0.00	284.00	284.00	284.00	**0.00**	0.00
NSF212	522.00	503.00	504.10	505.00	3.43	0.06	509.00	511.10	514.00	**2.09**	0.19
NSF23	285.00	285.00	285.00	285.00	**0.00**	0.00	285.00	285.00	285.00	**0.00**	0.00
NSF248	505.00	486.00	486.40	487.00	3.68	0.05	490.00	492.50	495.00	**2.48**	0.16
NSF3	285.00	285.00	285.00	285.00	**0.00**	0.00	285.00	285.00	285.00	**0.00**	0.00
NSF48	469.00	461.00	462.00	463.00	1.49	0.07	464.00	466.30	467.00	**0.58**	0.10
SUN	722.00	643.00	643.30	644.00	10.90	0.05	646.00	646.00	646.00	**10.53**	0.00
Average					4.18					3.14	

The smallest optimality gap for each instance is displayed in boldface

MS was 9.72 %, while that for BRKGA was only 8.23 %. Relating to set C, BRKGA found better solutions than MS for all, but the six instances where both heuristics found the optimal solution value. The average optimality gap of BRKGA for set C was only 3.14 %, while that for MS was 4.18 %. The maximum gap for MS was 10.90 %, while that for BRKGA was 10.53 %. To summarize, BRKGA found better solution values and smaller maximum and average optimality gaps for all test sets. In addition, the coefficient of variation of BRKGA was at most 1.78 % from the average (observed for network ATT2 in set C) and less than 1 % for 68 out of the 72 test instances.

Run time distributions or time-to-target plots display on the ordinate axis the probability that an algorithm will find a solution at least as good as a given target value within a given running time, shown on the abscissa axis. Run time distributions have also been advocated by Hoos and Stützle [22] as a way to characterize the running times of stochastic local search algorithms for combinatorial optimization. In the next experiment, both heuristics MS and BRKGA were made to stop whenever a solution with cost smaller than or equal to a given target value was found. This target value was set to the average cost of the solutions obtained by MS in the previous experiments, i.e., the average of ten 10-min runs of MS for each instance. The heuristics were run 200 times on each instance, with different seeds for the pseudo-random number generator. Next, the empirical probability distributions of the time taken by each heuristic to find a target solution value are plotted. To plot the empirical distribution for each heuristic, we followed the methodology described in [1,2]. We associate a probability $p_i = (i - \frac{1}{2})/200$ with the i-th smallest running time t_i and plot the points $z_i = (t_i, p_i)$, for $i = 1, \ldots, 200$. The more to the left is a plot, the better is the algorithm corresponding to it.

We illustrate the runtime distributions (or ttt-plots, for short) for one instance of each test set. The runtime distributions for the instance of set A defined by network ATT2 are shown in Fig. 6. BRKGA finds solutions with 856 accepted lightpaths with a probability close to 100 % in less than 10 s of running time, while MS may take up to 2500 s to find solutions as good as them with the same probability. Similar results were observed for the instances of sets B and C. The ttt-plots for the instance of set B defined by network FRANCE are shown in Fig. 7. BRKGA finds solutions with 1072 accepted lightpaths with a probability close to

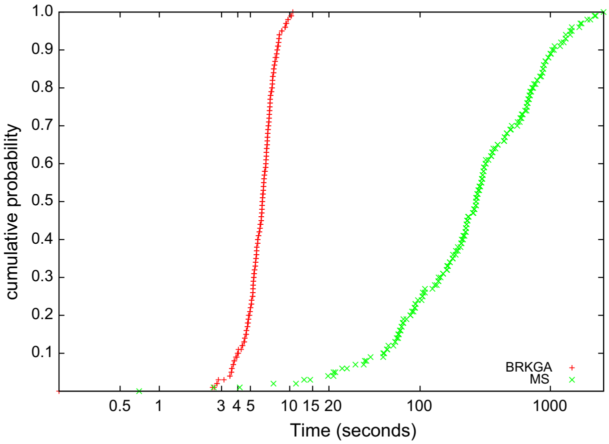

Fig. 6 Runtime distributions for the instance of set A defined by network ATT2 with the target value set at 856: $P(T_{BRKGA} \leq T_{MS}) = 0.985$.

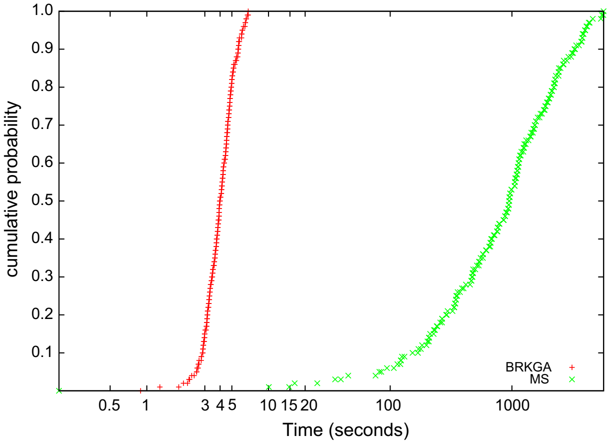

Fig. 7 Runtime distributions for the instance of set B defined by network FRANCE with the target value set at 1072: $P(T_{BRKGA} \leq T_{MS}) = 0.995$.

100 % in less than 9 s of running time, while MS may take up to 5000 s to find solutions as good as them with the same probability. Finally, the runtime distributions for the instance of set C defined by the network DFN-GWIN is shown in Fig. 8. In this case, BRKGA finds solutions with 1944 accepted lightpaths with a probability close to 100 % in less than 5 s of running time, while MS may take up to 1500 s to find solutions as good as them with the same probability.

We also used the tool proposed by Ribeiro et al. [51] to perform a direct numerical comparison of heuristics MS and BRKGA. Let T_{BRKGA} (resp. T_{MS}) be the continuous random variable representing the time needed by heuristic BRKGA (resp. MS) to find a solution as good as a given target value and let $P(T_{BRKGA} \leq T_{MS})$ be the probability that BRKGA converges faster than MS, computed by the software available in [50]. The captions of Figs. 6, 7 and 8 show that $P(T_{BRKGA} \leq T_{MS}) \geq 0.985$ for the above three instances, further illustrating the superiority of BRKGA with respect to MS.

Figures 9 and 10 illustrate the evolution of the solution population along 200 generations of BRKGA for one execution of instances DFN-GWIN and ATT2. They show that the biased random-key genetic algorithm is able to continuously evolve the solution population and to improve the best solution value.

Finally, Figs. 11 and 12 illustrate (for the instance of set B defined by network DFN-GWIN with 20 lightpath requests) how the best solutions found by BRKGA and by the multi-start procedure evolve along the first six and the first 60 s of processing time, respectively. They show that the biased random-key genetic algorithm systematically finds better solutions faster than the other algorithm. The best solution obtained by BRKGA is better than that found by multi-start at any time along the run whose results are displayed in these figures. This same behavior illustrated for this run is observed for all test instances.

These results show that BRKGA identifies the best relationships between the keys and the good solutions throughout the evolutionary process, converging to high-quality, near-optimal solutions.

In the third experiment, we compare the quality of the solutions provided by BRKGA and the tabu search (TS) heuristic for the instance set D, which was used by Dzongang et al. [12] to evaluate the performance of their heuristic. Upper bounds for the 30 instances are

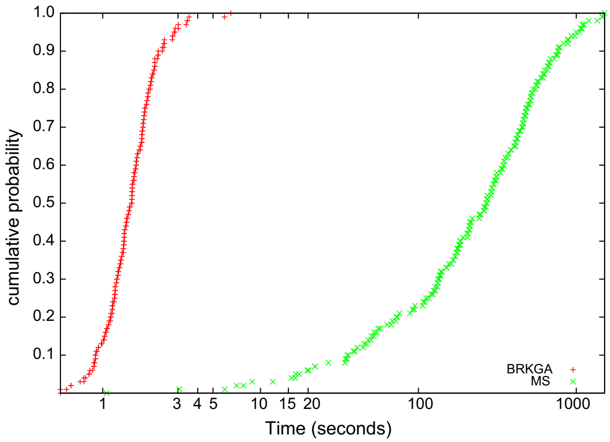

Fig. 8 Runtime distributions for the instance of set C defined by network DFN-GWIN with the target value set at 1944: $P(T_{BRKGA} \leq T_{MS}) = 0.996$.

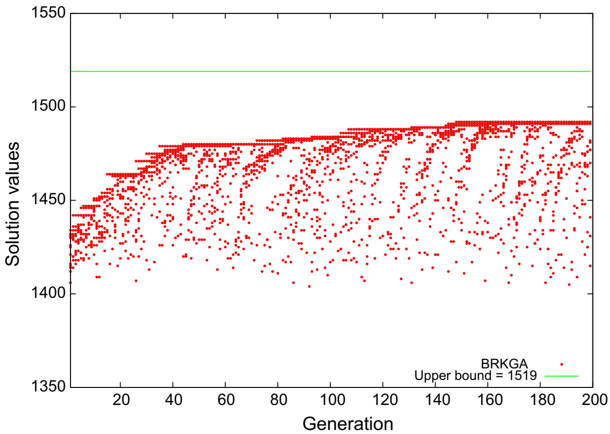

Fig. 9 Population evolution for the instance of set B defined by network DFN-GWIN: the value of the best solution found by BRKGA after 4.42 s (200 generations) is 1492, while the best solution value after 10 min of running time is 1502

known from [33]. The tabu search heuristic was implemented in C++ and the experiments were performed in a Pentium 4 with 2.4 GHz. Individual running times for each instances are not reported. However, it is said that the tabu search run for up to 60 s. Our implementation of BRKGA was run 10 times for each instance, with different seeds for the random number generator [53]. The stopping criterion of BRKGA was set to 60 s, in order to make its running times compatible with those of the tabu search heuristic.

The results for this experiment are displayed in Table 5. The first column provides the number (λ) of wavelengths available for the instance. The next three columns display the upper bound, the optimality gap of TS, and the average optimality gap of BRKGA for the corresponding NSF instance with λ wavelengths. The last three columns show the upper bound, the optimality gap of TS, and the average optimality gap of BRKGA for the corresponding EONNET instance with λ wavelengths. The smallest optimality gap for each instance is

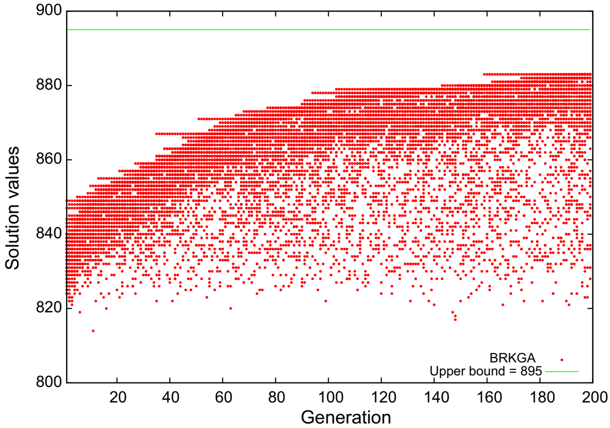

Fig. 10 Population evolution for the instance of set A defined by network ATT2 the value of the best solution found by BRKGA after 42.00 s (200 generations) is 883, while the best solution value after 10 min of running time is 889

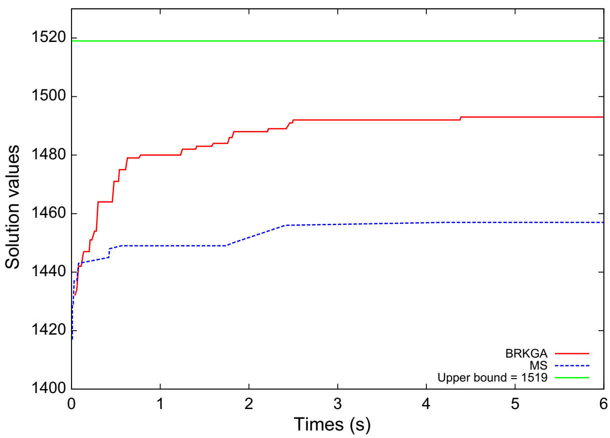

Fig. 11 Evolution of the best solutions found by BRKGA and MS along the six first seconds of processing time for the instance of set B defined by network DFN-GWIN with 20 lightpaths: the best solution value obtained by BRKGA is 1493, while that found by MS is only 1457

displayed in boldface. Regarding the 15 instances with 14 nodes and 21 links based on NSF, TS found smaller optimality gaps on 8 instances, while BRKGA out performs TS on five instances. The average optimality gap of TS and BRKGA was 1.41 and 2.50 %, respectively. However, BRKGA found optimal solutions for six (out of the 15) instances, while TS found optimal solutions for only two instances. Regarding the 15 largest instances with 20 nodes and 39 links based on EONNET, BRKGA found solutions better than those of TS for 13 instances, while both heuristics found the optimal solution for the other two instances. The average optimality gap of TS and BRKGA was 1.53 and 0.62 %, respectively. For the 30 instances in this set together, The average optimality gap of TS and BRKGA was 1.47 and 1.56 %, respectively. These results show that BRKGA finds solutions competitive with those of the tabu search heuristic of [12].

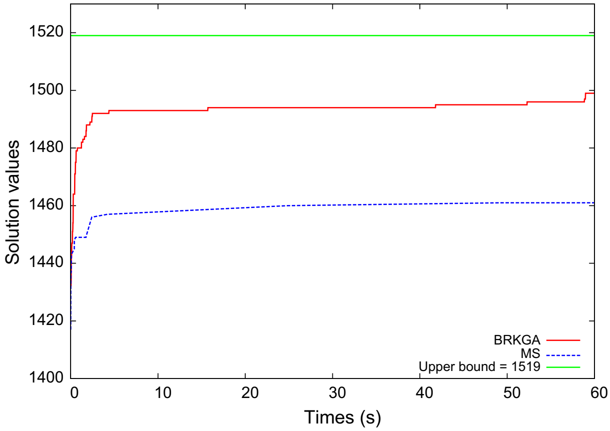

Fig. 12 Evolution of the best solution values found by BRKGA and MS along the 60 first seconds of processing time for the instance of set B defined by network DFN-GWIN with 20 lightpaths: the best solution value obtained by BRKGA is 1499, while that found by MS is only 1461

Table 5 BRKGA versus the tabu search (TS) of [12] for instances of test set D

λ	NSF			EONNET		
	UB	TS (%)	BRKGA (%)	*UB*	TS (%)	BRKGA (%)
10	198	**0.51**	4.65	285	1.40	**1.23**
11	218	**4.59**	8.49	301	2.33	**1.43**
12	218	**0.00**	3.62	317	3.15	**1.58**
13	238	**4.20**	7.65	329	3.34	**1.40**
14	238	**0.42**	2.98	337	2.67	**0.86**
15	248	**1.21**	3.27	344	1.74	**0.58**
16	258	**2.33**	3.53	350	1.43	**0.49**
17	263	**1.90**	2.05	356	1.12	**0.53**
18	267	2.25	**1.24**	362	1.66	**0.47**
19	268	1.49	**0.00**	367	1.63	**0.19**
20	268	1.12	**0.00**	370	1.08	**0.22**
21	268	0.75	**0.00**	373	0.80	**0.29**
22	268	0.37	**0.00**	374	0.53	**0.03**
23	268	**0.00**	**0.00**	374	**0.00**	**0.00**
24	268	**0.00**	**0.00**	374	**0.00**	**0.00**
Average:		1.41	2.50		1.53	0.62

The smallest optimality gap for each instance is displayed in boldface

6 Conclusions

We have shown that most of the work in the literature about the max-RWA version of the problem of routing and wavelength assignment is based on integer programming formulations and focus into finding exact solutions or upper bounds to the optimal value. Only small instances with no more than 27 nodes are solved to optimality, and only upper bounds are known for larger instances with up to 90 nodes in sets A, B, and C.

We proposed new greedy constructive heuristics and a biased random-key genetic algorithm. Computational experiments showed that the greedy heuristic SPT outperformed the best greedy heuristics in literature. In addition, the biased random-key genetic algorithm using SPT as the decoding heuristic found near-optimal solutions with average optimality gaps of 3.54, 3.99, 3.14, and 1.56 % for the instances in set A, B, C, and D, respectively. The optimality gap of this heuristic was at most 10.53 % over all 102 test instances, observed for network SUN in test set C.

References

1. Aiex, R., Resende, M., Ribeiro, C.C.: Probability distribution of solution time in GRASP: an experimental investigation. J. Heuristics **8**, 343–373 (2002)
2. Aiex, R.M., Resende, M.G.C., Ribeiro, C.C.: TTTPLOTS: a perl program to create time-to-target plots. Optim. Lett. **1**, 355–366 (2007)
3. Banerjee, D., Mukherjee, B.: A practical approach for routing and wavelength assignment in large wavelength-routed optical networks. IEEE J. Sel. Areas Commun. **14**, 903–908 (1996)
4. Bean, J.C.: Genetic algorithms and random keys for sequencing and optimization. ORSA J. Comput. **2**, 154–160 (1994)
5. Belgacem, L., Puech, N.: Solving large size instances of the RWA problem using graph partitioning. In: International Conference on Optical Network Design and Modeling, IEEE, Barcelona (2008)
6. Brandão, J.S., Noronha, T.F., Resende, M.G.C., Ribeiro, C.C.: A biased random-key genetic algorithm for single-round divisible load scheduling. Int. Trans. Oper. Res. **22**, 823–839 (2015)
7. Buriol, L.S., Resende, M.G.C., Ribeiro, C.C., Thorup, M.: A hybrid genetic algorithm for the weight setting problem in OSPF/IS–IS routing. Networks **46**, 36–56 (2005)
8. Buriol, L.S., Resende, M.G.C., Thorup, M.: Survivable IP network design with OSPF routing. Networks **49**, 51–64 (2007)
9. Buriol, L.S., Hirsch, M.J., Pardalos, P.M., Querido, T., Resende, M.G.C., Ritt, M.: A biased random-key genetic algorithm for road congestion minimization. Optim. Lett. **4**, 619–633 (2010)
10. Chen, C., Banerjee, S.: A new model for optimal routing and wavelength assignment in wavelength division multiplexed optical networks. In: Proceedings of the Fifteenth Annual Joint Conference of the IEEE Computer Societies. Networking the Next Generation, vol. 1, pp. 64–171. San Francisco (1996)
11. Choi, J.S., Golmie, N., Lapeyrere, F., Mouveaux, F., Su, D.: A functional classification of routing and wavelength assignment schemes in DWDM networks: Static case. In: Proceedings of the 7th International Conference on Optical Communication and Networks, pp. 1109–1115. Paris (2000)
12. Dzongang, C., Galinier, P., Pierre, S.: A tabu search heuristic for the routing and wavelength assignment problem in optical networks. IEEE Commun. Lett. **9**, 426–428 (2005)
13. Ericsson, M., Resende, M.G.C., Pardalos, P.M.: A genetic algorithm for the weight setting problem in OSPF routing. J. Comb. Optim. **6**, 299–333 (2002)
14. Erlebach, T., Jansen, K.: The complexity of path coloring and call scheduling. Theor. Comput. Sci. **255**, 33–50 (2001)
15. Gonçalves, J.F., de Magalhães Mendes, J.J., Resende, M.G.C.: A hybrid genetic algorithm for the job shop scheduling problem. Eur. J. Oper. Res. **167**, 77–95 (2005)
16. Gonçalves, J.F., Mendes, J.J.M.: A hybrid genetic algorithm for assembly line balancing. J. Heuristics **8**, 629–642 (2002)
17. Gonçalves, J.F., Resende, M.G.C.: An evolutionary algorithm for manufacturing cell formation. Comput. Ind. Eng. **47**, 247–273 (2004)
18. Gonçalves, J.F., Resende, M.G.C.: Biased random-key genetic algorithms for combinatorial optimization. J. Heuristics **17**, 487–525 (2011)
19. Goulart, N., de Souza, S.R., Dias, L.G.S., Noronha, T.F.: Biased random-key genetic algorithm for fiber installation in optical network optimization. In: Proceedings of the 2011 IEEE Congress on Evolutionary Computation, pp. 2267–2271. New Orleans (2011)
20. Graham, R.L.: Bounds on multiprocessing timing anomalies. SIAM J. Appl. Math. **17**, 416–429 (1969)
21. Grötschel, M., Lovász, L., Schrijver, A.: Geometric Algorithms and Combinatorial Optimization. Algorithms and Combinatorics, vol. 2. Springer, Berlin (1993)

22. Hoos, H., Stützle, T.: Evaluation of Las Vegas algorithms - Pitfalls and remedies. In: Cooper, G., Moral, S. (eds.) Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence, pp. 238–245. Madison (1998)
23. Hou, E.S., Ansari, N., Ren, H.: A genetic algorithm for multiprocessor scheduling. IEEE Trans. Parallel Distrib. Syst. **5**(2), 113–120 (1994)
24. Hyytiä, E., Virtamo, J.: Wavelength assignment and routing in WDM networks. In: Fourteenth Nordic Teletraffic Seminar, pp. 31–40. Copenhagen (1998)
25. Hyytiä, E., Virtamo, J.: Wavelength assignment in multifibre in WDM-networks. Technical report COST257TD(99)04, Helsinki University of Technology (1999)
26. Jaumard, B., Meyer, C., Thiongane, B., Yu, X.: ILP formulations and optimal solutions for the RWA problem. In: Proceedings of the IEEE Global Telecommunications Conference, vol. 3, pp. 1918–1924. Dallas (2004)
27. Jaumard, B., Meyer, C., Thiongane, B.: ILP formulations for the routing and wavelength assignment problem: symmetric systems. In: Handbook of Optimization in Telecommunications, pp. 637–677. Springer, Berlin (2006)
28. Jaumard, B., Meyer, C., Yu, X.: How much wavelength conversion allows a reduction in the blocking rate? J. Opt. Netw. **5**, 81–900 (2006)
29. Jaumard, B., Meyer, C., Thiongane, B.: Comparison of ILP formulations for the RWA problem. Opt. Switch. Netw. **4**, 157–172 (2007)
30. Jaumard, B., Meyer, C., Thiongane, B.: On column generation formulations for the RWA problem. Discrete Appl. Math. **157**, 1291–1308 (2009)
31. Johnson, D.S., Demers, A., Ullman, J.D., Garey, M.R., Graham, R.L.: Worst-case performance bounds for simple one-dimensional packing algorithms. SIAM J. Comput. **3**, 299–325 (1974)
32. Kleinberg, J.: Approximation algorithms for disjoint paths problems. Ph.D. thesis, MIT, Cambridge (1996)
33. Krishnaswamy, R., Sivarajan, K.: Algorithms for routing and wavelength assignment based on solutions of LP-relaxation. IEEE Commun. Lett. **5**, 435–437 (2001)
34. Lee, T., Lee, K., Park, S.: Optimal routing and wavelength assignment in WDM ring networks. IEEE J. Sel. Areas Commun. **18**, 2146–2154 (2000)
35. Lei, D.: Fuzzy job shop scheduling problem with availability constraints. Comput. Ind. Eng. **58**, 610–617 (2010)
36. Li, G., Simha, R.: The partition coloring problem and its application to wavelength routing and assignment. In: Proceedings of the First Workshop on Optical Networks, pp. 1–19. Dallas (2000)
37. Malve, S., Uzsoy, R.: A genetic algorithm for minimizing maximum lateness on parallel identical batch processing machines with dynamic job arrivals and incompatible job families. Comput. Oper. Res. **34**, 3016–3028 (2007)
38. Manohar, P., Manjunath, D., Shevgaonkar, R.K.: Routing and wavelength assignment in optical networks from edge disjoint path algorithms. IEEE Commun. Lett. **5**, 211–213 (2002)
39. Marković, G., Aćimović-Raspopović, V.: Solving the RWA problem in WDM optical networks using the BCO meta-heuristic. Telfor J. **2**, 43–48 (2010)
40. Martins, A.X., Duhamel, C., De Souza, M.C., Saldanha, R.R., Mahey, P.: A VND-ILS heuristic to solve the RWA problem. In: Pahl, J., Reiners, T., Vo, S. (eds.) Network Optimization, pp. 577–582. Springer, Berlin (2011)
41. Martins, A.X., Duhamel, C., Mahey, P., de Souza, M.C., Saldanha, R.R.: Geração de colunas para o problema de roteamento e atribuição de comprimentos de onda. In: Anais do XLIV Simp. Bras. Pesq. Operacional, pp. 1–12. Rio de Janeiro (2012)
42. Martins, A.X.: Metaheurísticas e formulações para a resolução do problema de roteamento e alocação de comprimentos de onda em redes ópticas. Ph.D. thesis, Universidade Federal de Minas Gerais (2011)
43. Martins, A.X., Duhamel, C., Mahey, P., Saldanha, R.R., de Souza, M.C.: Variable neighborhood descent with iterated local search for routing and wavelength assignment. Comput. Oper. Res. **39**, 2133–2141 (2012)
44. Noronha, T.F., Resende, M.G.C., Ribeiro, C.C.: Efficient implementation of heuristics for routing and wavelength assigment. Lect. Notes Comput. Sci. **5038**, 169–180 (2008)
45. Noronha, T.F., Resende, M.G.C., Ribeiro, C.C.: A biased random-key genetic algorithm for routing and wavelength assignment. J. Global Optim. **50**, 503–518 (2011)
46. Noronha, T.F., Ribeiro, C.C.: Routing and wavelength assignment by partition coloring. Eur. J. Oper. Res. **171**, 797–810 (2006)
47. Qin, H., Liu, Z., Zhang, S., Wen, A.: Routing and wavelength assignment based on genetic algorithm. Commun. Lett. **6**(10), 455–457 (2002)

48. Ramaswami, R., Sivarajan, K.: Optimal routing and wavelength assignment in all-optical networks. In: Proceedings of IEEE INFOCOM Conference on Computer Communications, pp. 970–979. Toronto (1994)
49. Reis, R., Ritt, M., Buriol, L.S., Resende, M.G.C.: A biased random-key genetic algorithm for OSPF and DEFT routing to minimize network congestion. Int. Trans. Oper. Res. **18**, 401–423 (2011)
50. Ribeiro, C.C., Rosseti, I.: tttplots-compare: a perl program to compare time-to-target plots or general runtime distributions of randomized algorithms. Optim. Lett. **9**, 601–614 (2015)
51. Ribeiro, C.C., Rosseti, I., Vallejos, R.: Exploiting run time distributions to compare sequential and parallel stochastic local search algorithms. J. Global Optim. **54**, 405–429 (2012)
52. Roque, L., Fontes, D., Fontes, F.: A hybrid biased random key genetic algorithm approach for the unit commitment problem. J. Comb. Optim. **28**(1), 140–166 (2014)
53. Schrage, L.: A more portable Fortran random number generator. ACM Trans. Math. Softw. **5**, 132–138 (1979)
54. Skorin-Kapov, N.: Routing and wavelength assigment in optical networks using bin packing based algorithms. Eur. J. Oper. Res. **177**, 1167–1179 (2007)
55. Spears, W., deJong, K.: On the virtues of parameterized uniform crossover. In: Belew, R., Booker, L. (eds.) Proceedings of the Fourth International Conference on Genetic Algorithms, pp. 230–236. Morgan Kaufman, San Mateo (1991)
56. Wang, C.S., Uzsoy, R.: A genetic algorithm to minimize maximum lateness on a batch processing machine. Comput. Oper. Res. **29**, 1621–1640 (2002)
57. Zang, H., Jue, J.P., Mukherjee, B.: A review of routing and wavelength assignment approaches for wavelength-routed optical WDM networks. Opt. Netw. Mag. **1**, 47–60 (2000)
58. Zheng, J.N., Chien, C.F., Gen, M.: Multi-objective multi-population biased random-key genetic algorithm for the 3-D container loading problem. Comput. Ind. Eng. **89**, 80–87 (2015)