© International Journal of Combinatorial Optimization Problems and Informatics, Vol. 3, No. 1, Jan-April 2012, pp. 22-37. ISSN: 2007-1558.

Cutting path as a Rural Postman Problem: solutions by Memetic Algorithms

Ana Maria Rodrigues
Instituto de Engenharia de Sistemas e Computadores do Porto
Instituto Superior de Contabilidade e Administração do Porto,
Porto, Portugal
amr@inescporto.pt

José Soeiro Ferreira
Faculdade de Engenharia, Universidade do Porto
Instituto de Engenharia de Sistemas e Computadores do Porto
Porto, Portugal
jsoeiro@inescporto.pt

Abstract. The Rural Postman Problem (RPP) is a particular Arc Routing Problem (ARP) which consists of determining a minimum cost circuit on a graph so that a given subset of required edges is traversed. The RPP is an NP-hard problem with significant real-life applications. This paper introduces an original approach based on Memetic Algorithms - the MARP algorithm - to solve the RPP and, also deals with an interesting Industrial Application, which focuses on the path optimization for component cutting operations. Memetic Algorithms are a class of Metaheuristics which may be seen as a population strategy that involves cooperation and competition processes between population elements and integrates "social knowledge", using a local search procedure. The MARP algorithm is tested with different groups of instances and the results are compared with those gathered from other publications. MARP is also used in the context of various real-life applications.

Keywords: Cutting Path Application, Rural Postman Problem, Memetic Algorithms.

1 Introduction

Many industries need to determine good layouts and path planning to cut pieces using various cutting tools and procedures that are appropriate for the materials in use. This work is motivated by specific continuous process path-cutting applications, meaning that the cutting tool never leaves the cutting surface, and there are no restrictions with regard to completely cutting a piece after initiation. These applications are modeled as Rural Postman Problems (RPP) and solved as such. Furthermore, an original method based on Memetic Algorithms (MA) **-** MARP algorithm **-** is introduced and used to solve the RPP. Therefore, the paper is twofold:

1. A new algorithm for the RPP is presented and tested;
2. Potential applications to path cutting planning are illustrated.

The RPP is more general than the Chinese Postman Problem (CPP), a well-known Arc Routing Problems (ARP), which consists of finding the shortest circuit that traverses each edge of a graph at least once. These edges can be directed, undirected or both. In the case of RPP, not all of the edges have to be traversed by the circuit, only those included in a specified subset of required edges. The RPP is applied to a variety of practical contexts which include mail delivery, garbage collection (Ghiani et al. [1] present a case study modeled as a particular ARP), street cleaning, road gritting, meter reading and laser plotter applications [2, 3,4].

Received Sep 5, 2011 / Accepted Oct 3, 2011

Undirected, directed and mixed RPP are NP-hard [5,6] which helps to explain the existence of dedicated heuristics and the research on new approaches that may be based on Metaheuristics. MA were selected to be used in this paper in order to obtain high quality solutions in a short period of time and bearing in mind its industrial applications. Furthermore, it is the first time that MA have been employed in such a context and thus represents a good opportunity to evaluate their efficacy. MA are population-based Metaheuristic which combines local search heuristics with crossover operators. They aim to integrate "knowledge", using the local search procedure, in order to complement the "genetic information" acquired by the crossover operators. Consequently, there is an expectation for an improvement in its performance for difficult combinatorial optimization problems or problems with large instances.

The industrial cutting path applications, modeled as an RPP, are presented in Subsection 2.1. In Subsection 2.2, the RPP is defined in connection with other ARP. MA and MARP are explained in Section 3. Section 4 contains the computational results and the evaluation of the MARP algorithm, both for solving RPP and its applications to path cutting. Finally, some conclusions are drawn in the last section, Section 5.

2 Cutting path as Rural Postman Problem

Industrial applications dealing with the path optimization for component cutting operations are presented next. As these applications may be modeled as RPP and solved as such, the Subsection 2.2 is dedicated to a description of these problems.

2.1 Cutting path planning application

The need to cut materials using different tools, such as flame or electrified cutters, artificial diamonds or water pressure is frequent in various different industries. Specific publications related to this subject can be highlighted: Manber and Israni [7] developed three algorithms for a manufacturing situation related to the generation of a sequence of torch paths, given a nesting of parts on a stock sheet and the bridges to connect them. Various practical constraints are considered and the objective is to minimize the number of pierce points. Ghiani and Improta [8] presented a model to solve a laser-plotter beam routing problem as a constrained arc routing problem; Moreira et al. [9] provided two heuristics based on node routing for a complex cutting application in the metallomechanical industry, after devising the Dynamic RPP; Imahori et al. [10] also dealt with the generation of path cutting for hard materials - a geometrical heuristic approach is presented which facilitates the representation of complicated conditions resulting from the specific application; Huang et al. [11] present a two-stage methodology for flame cutting rectangular pieces - the pieces are grouped into larger rectangles to minimize the distance travelled within blocks and a genetic algorithm determines multiple starting points for each block. Usberti et al. [12] present the Open Capacitated Arc Routing Problem, where tours are not constrained to form circles, and they also present some applications, such as path cutting determination problem.

Each industrial application has its own specific characteristics and restrictions. This work is motivated, as mentioned previously, by specific path planning applications involving a continuous cutting procedure. The diameter of the cutting tool is not significant and so pieces can touch each other, which is important to minimize waste and reduce the cutting path. Furthermore, the cut out pieces should not be allowed to fall, instead they should remain in position, or be removed from the cutting surface. This last facet is a fundamental difference in relation to the problem tackled in Moreira et al. [9], also meaning that there is no advantage in using the concept of dynamic graphs.

To illustrate the situation, Fig. 1 depicts a practical example consisting of a circular plate with 28 components that must be cut out using a cutting tool that does not leave the cutting surface until all the pieces have been completely cut out. The example is solved by the MARP algorithm (http://www.inescporto.pt/amr).

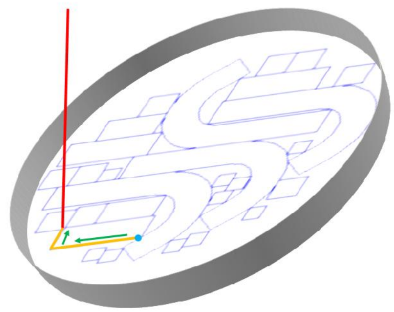


Fig. 1. Circular plate and cutting tool.

The whole problem is frequently separated into two stages: packing/nesting and cutting. Another intermediate phase to define arbitrary connections between the pieces will also be included in this paper (a requirement of the continuous cutting process):

1. Nesting - this process involves the layout of the small pieces over the plate while trying to minimize waste;
2. Bridges - bridges must be launched between the pieces, so that the cutting tool can move from one piece to another, without cutting the interior of any piece (see Fig. 2);
3. Cutting - this process involves cutting all of the pieces using the shortest path (a consequence is the minimization of the cutting time, which is an important goal).

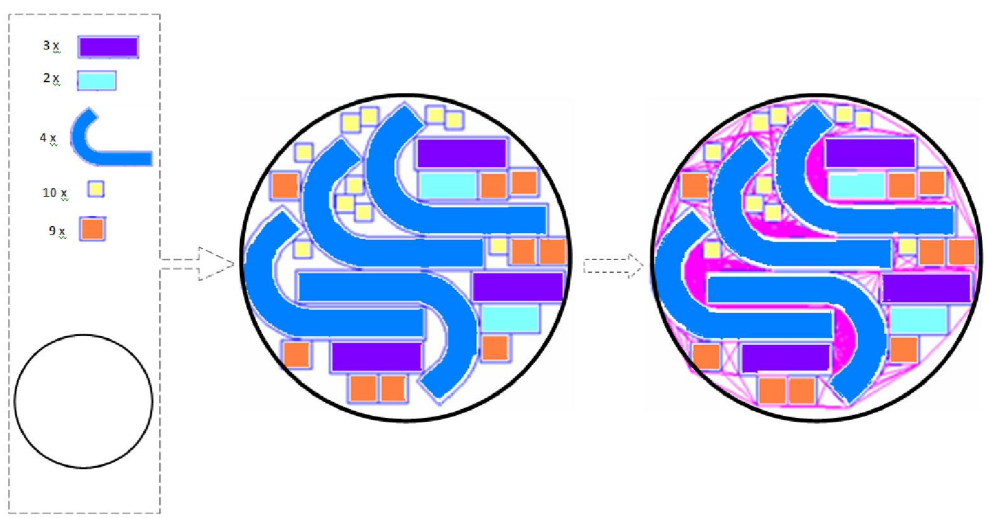


Fig. 2. Nesting and Bridges launching.

There is no special difference for planning purposes between cutting one edge of a piece, one bridge or, potentially, a plate portion that does not correspond to one effective cut. What is most important is the reduction of the distance/time travelled by the cutting tool. There are no restrictions in terms of completely cutting a piece after initiation - it can be abandoned and revisited later on to finish the cut.

This paper will only refer to the cutting phase. References to the other phases include, Moreira et al. [9] for bridges launching and shape representation, and Gomes and Oliveira [13], Oliveira et al. [14], Burke et al. [15] and Bennell and Oliveira [16] for manipulation.

Fig. 3 illustrates the proposal of modeling these cutting problem as a Rural Postman Problem (RPP), which is defined in the next section. These ARP are specific because they include both required and non-required edges - required edges must be crossed at least once. The edges of the pieces (P1 and P2) that will be cut will act as the required edges, while the three segments between the pieces (launched bridges) are the non required or facultative edges (some will be chosen to move between the pieces).

This last step is not necessary for the envisaged industrial applications, because their correspondent graphs should only have one connected component; the graph with only the required edges is connected.

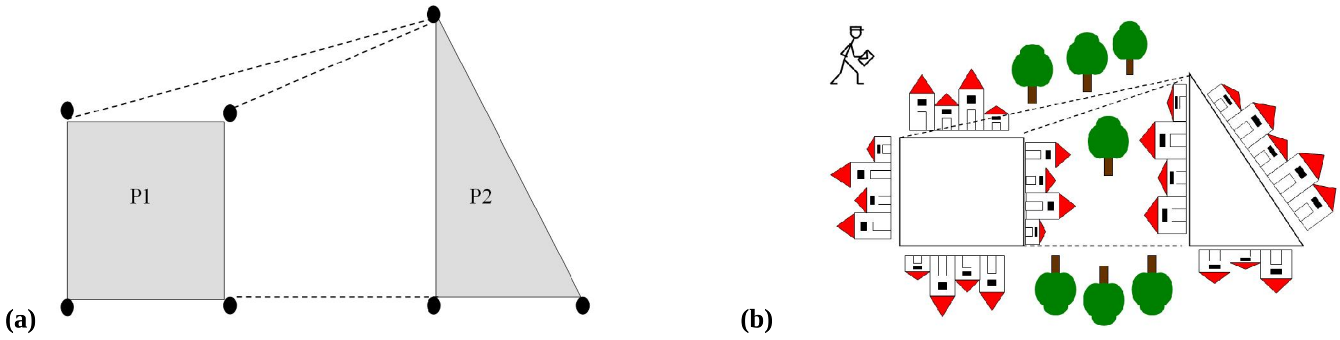


Fig. 3. Cutting Problem **(a)** modeled as an RPP **(b)**.

2.2 Rural Postman Problem

An ARP consists of determining a traversal of the lowest cost of the set (or subset) of edges on a graph, $G=(V,E)$, V is the set of vertices and E is the set of edges. The edges can be directed, undirected or both. This paper is only concerned with undirected edges.

There are plenty of applications of this kind to problems in many different areas. Perhaps the first reference to an ARP is the famous Königsberg bridge problem [17]. Nowadays this problem appears in a large variety of practical contexts such as: mail delivery, delivery of telephone books, garbage collection, street sweepers, road gritting, inspection of streets for maintenance, meter reading, snow removal, school bus route, planning internet routing and manufacturing printed circuit boards.

The Chinese Postman Problem (CPP) is an ARP where the aim is to find a minimum cost closed path traversing every edge $e \in E$ of the graph G with a set of vertices V, **at least once**. A cost $c: E \longrightarrow IR_0^+$ is associated with each edge. The undirected case can be solved in polynomial time. However, there are some extensions of the CPP which are NP-hard problems, such as the Windy Postman Problem, the Capacitated Chinese Postman Problem and the Hierarchical Postman Problem, see Laporte and Osman [18]. One of these extensions is the Rural Postman Problem (RPP), originally presented in Orloff [19] and described as follows. The RPP consists of finding the minimum closed path of G so that every edge in E_R is traversed at least once given a set of required edges $E_R \subseteq E$. If $E_R=E$, the RPP is reduced to the CPP.

Integer linear programming formulations have been proposed for the undirected RPP, the first formulation, suggested by Christofides et al. [20]. Exact methods are proposed in articles such as Christofides et al. [20], Corberán and Sanchis [21], Ghiani and Laporte [22] or Fernández et al. [23]. Constructive heuristics are also frequent, such as those presented in Frederickson [24] and Pearn and Wu [25]. Metaheuristics such as Ant Colony Optimization are proposed in Laganà et al. [26]. Significant results were obtained in Hertz et al. [27] with heuristics, in Ghiani and Laporte [22] using branch and cut and also in Fernández et al. [23], where the formulation introduced in [28] was modified to improve its computational viability. The heuristic for the Undirected RPP presented in Frederickson [24] is classified by different researchers as the best known constructive heuristic. More references in this field are related to the Monte Carlo Method in Córdoba et al. [29] that was used to simulate a vehicle in a graph randomly traveling and to simulate the Tabu Search implementation in Corberán et al. [30]. A Tabu Search method, in conjunction with Frederickson's heuristic (for solving the RPP), is also used in Groves et al. [31] to solve a problem that involves routing and scheduling at the same time. The RPP is addressed in generic publications on ARP, such as Dror [32], Eiselt et al. [4] and, more recently, the relevant survey and annotated bibliography of Corberán and Prins [33] and Rodrigues and Ferreira [34]. There are many applications for the RPP [18], as highlighted in the Introduction and in the previous Subsection.

3 MARP – a Memetic Algorithm for the RPP

MA are a class of Metaheuristics which has been evolving since the end of the 1980s. They can be interpreted as a cooperative-competitive strategy to optimize agents [35]. Their implementations are supported by a population-based search which aims to use all available knowledge on the problem under study. Following some form of recombination, to exchange information on the solutions, the general aim is to improve these solutions. This may be achieved through the combination of local searches as implemented in this paper, for instance.

Some authors have found connections between MA and Genetic Algorithms (GA). GA began being researched in the 1970's by John Holland and they constitute a computational model of biological evolution [36]. Norways, the basis of these algorithms is well-known. A reproductive process selects solutions (parents) within a population of solutions in order to produce other solutions (children) which have some of the characteristics of each progenitor [37]. In nature, the strongest individuals survive and the same occurs in GA. In fact, the strongest solutions are preserved. The success of GA essentially comes from its simplicity and efficacy in finding good solutions in short computational times and also from new developments such as ‘hybrid genetic algorithms’. MA may also be seen as a class of population-based algorithms that generalize the hybrid genetic algorithms. The latter combine GA strength and local search processes [38].

The term ‘Memetic Algorithm’ was introduced to suggest that ‘Cultural Evolution’ could be a better working metaphor to escape from biologically constrained thinking (see Norman and Moscato [35]). ‘Memetic’ is derived from the word ‘meme’ from R. Dawkins and its relationship with MA is the same as the relationship between ‘genes’ and GA. This means that the ‘meme’ is a ‘gene’ in the field of cultural evolution.

MA could thus be understood as strategies between agents (individuals of the population) that consist of competition and cooperation processes [35].

The implementation of MA in this paper (MARP algorithm) will follow the common scheme of adding a local optimizer. Just as when applying GA, new solutions emerge after recombination and mutation although, sometimes, these solutions-children are not as good as their progenitors. That is why a local search could help a child improve their fitness before becoming part of the population. The idea is to have a population of individuals where each one of them is an element of the local optima space.
Some features of a good MA implementation are presented in Buriol et al. [39]:

1. suitable recombination and mutation operators, such as in any GA;
2. an effective and fast local search algorithm, which is crucial in MA;
3. hierarchically structured population - some studies show that results are good when the implementation is within a hierarchical structure (See Berreta and Moscato [40]);
4. advanced data structures and intelligent codification mechanisms.

Several population structures have been explored by different authors. Pablo Moscato [41] presents a complete ternary tree with 13 nodes (also with 40, 121, etc.) that is applied to parallel machine scheduling problems. Other authors, such as Berretta and Rodrigues [42], use this tree structure of 13 nodes for a multistage capacitated lot-sizing problem. Ring structures have also been used in Moscato and Norman [43] where individuals are arranged on a ring, competing and cooperating. MA has appeared in other applications, such as Knapsack Problems [44], the Travelling Salesperson Problem [43] or the Periodic Capacitated Arc Routing Problem [45]. Cotta [46] gives an overview of generic Memetic Algorithms, and Moscato and Cotta [47] present an overview of MA with historical notes, applications and current developments in this area.

Before proceeding to the idea and explanation of MARP (next section), the necessary operations and graph transformations must be mentioned in order to obtain initial solutions:

1. Changing the graph by eliminating vertices that only support facultative edges, facultative edges that exceed the minimum cost between those vertices and repeated edges with the same cost (a graph simplification);
2. Transforming the vertices into even degree vertices (pairs of vertices with odd degree are randomly chosen and a shortest path between each pair is added);
3. Creating the minimum spanning tree.

Step 3 is not necessary for the envisaged industrial applications, because their correspondent graphs should only have one connected component, as mentioned in the previous section. Its inclusion was made to generalize the algorithm (for more than one connected component) and to make the computational tests possible. In order to create the minimum spanning tree, an algorithm was used that selects the best choice for each step to find an optimal solution, instead of considering all of the sequences of steps: a Greedy Algorithm with some adaptations in order to maintain the solutions' feasibility.

3.1 Population Structure

A ternary tree with three levels was chosen to structure the population, see Fig.4. This structure has been used before, for instance in Buriol et al. [39].

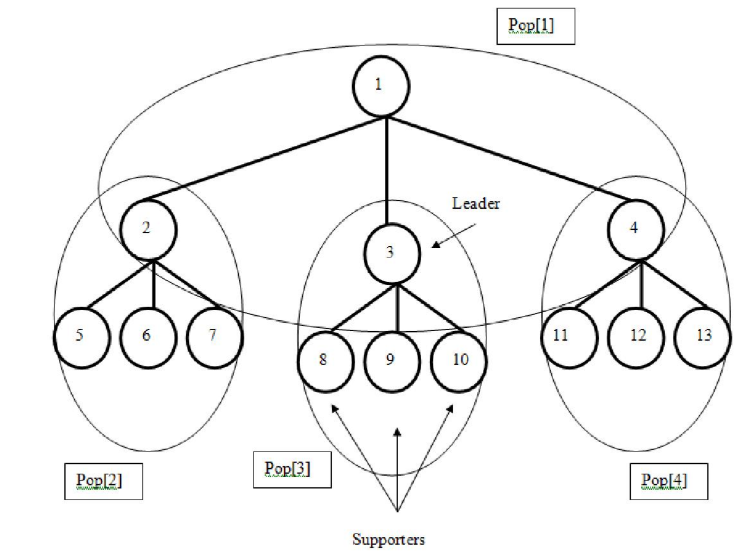


Fig. 4. Population Structure.

The population *P* is set up by 13 individuals hereinafter referred to as 'agents'. *P* is divided into 4 sub-populations, *Pop[i]*, *i=1,2,3,4*, each one consisting of one leader and three supporters. There are agents that simultaneously belong to 2 sub-populations, at the same time. The leader is always one level above its supporters. For instance, agent 3 is the leader of agents 8, 9 and 10 and belongs to *Pop[1]* and *Pop[3]* (see Fig. 4). Each agent of the population will handle two feasible solutions simultaneously. One is the best solution that the agent has found until that moment, *Pocket Solution (PS)*, while the other one is the current solution, *Current Solution (CS)*.

Before describing MARP, it is important to mention that Disjkstra's algorithm was used every time a minimum distance between two vertices needed to be calculated.

3.2 Structure Operation – Keeping Tree Structure

MARP´s good results can be due to the organization of the population - a ternary tree. Each of the population´s 13 agents, say *i*, owns 2 solutions, at each moment: *PS(i)* and *CS(i)*, with costs *PC(i)* and *CC(i)*, respectively. As explained before, the population is divided into four sub-populations and each sub-population (*Pop[i]*, *i=1, 2, 3, 4*) is set up by one leader and three supporters. In order to keep the hierarchical structure there are three crucial operations that must be taken into account [39]:

1. **UpdatePocket -** whenever $CC(i)<PC(i)$, for any *i* ($i=1, 2,...,13$), the solutions are switched;
2. **OrderChildren -** the supporters of one leader *i*, ($i=1, 2, 3, 4$) are in an ascending order of their Pocket Costs: $PC(i) \leq PC(3i-1) \leq PC(3i) \leq PC(3i+1)$, $\forall\ i \in \{1, 2, 3, 4\}$, where *i* is the leader of sub-population *Pop[i]*;
3. **PocketPropagation -** if the Pocket Cost of a supporter is better than the Pocket Cost of its leader, they exchange solutions.

3.3 Crossover Operation

Crossover is the process used to generate new individuals in the population. Each child presents the features that are common in both progenitors. The new solution (current solution) for the agent is found using Crossover between culturally related agents. In each iteration, the choice of the parents is carried out as follows:

CS(1)= Crossover(PS(2), PS(3));
CS(i)= Crossover(PS(3i), PS(3i+1)), for i=2, 3, 4;
CS(3i-1)= Crossover(PS(i), CS(3i)), for i=2, 3, 4;
CS(3i)= Crossover(PS(3i-1), CS(3i+1)), for i=2, 3, 4;
CS(3i+1)= Crossover(PS(3i), CS(3i-1)), for i=2, 3, 4.

Therefore, this guarantees the adequacy of the new solution.

3.4 Mutation Operation

Mutation's main goal is to prevent all of the members of the population becoming similar following some iterations. The computational experiments actually provided this evidence and so this operation has only been employed in order to introduce diversity among the agents.

Two parameters are related to Mutation:

1. *k* (operation frequency): the Mutation process begins following the *k* iterations without any improvement on the *PS* of an agent;
2. *d-opt* (the level of the Mutation): *d* edges, non-required or required and repeated edges are removed from the solution.

All agents can be mutated, except for Agent 1. This procedure guarantees that the best solutions obtained thus far will never be lost.

3.5 Local Search

Local search is an important phase of MA. Although the inspiration for the Crossover and Mutation operations comes from other MA implementations - but not the RPP - the Local Search procedure presented here is completely new. In this implementation, each agent attempts to find, a better solution individually. The explanation for this implementation is combined with an example described in Fig. 5 to Fig. 9. The graph $G^*=(V^*,E^*)$ (see Fig. 5) represents a solution (an Eulerian Graph with cost *y*) and $G=(V,E)$ (see Fig. 6) is the initial graph (with all edges):

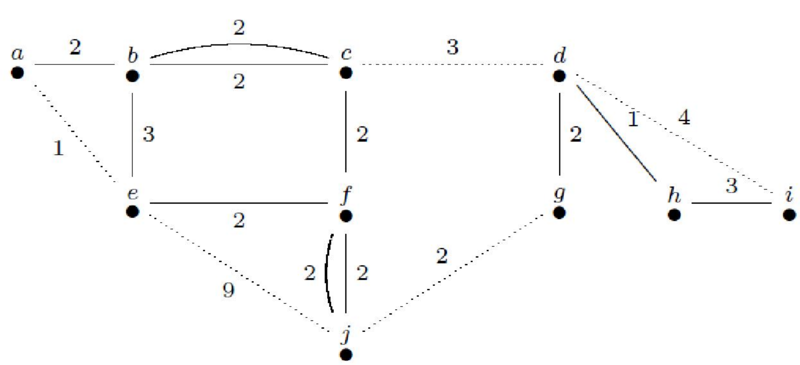


Fig. 5. An Eulerian Graph $G^*=(V^*,E^*)$, with cost $y = 42$.

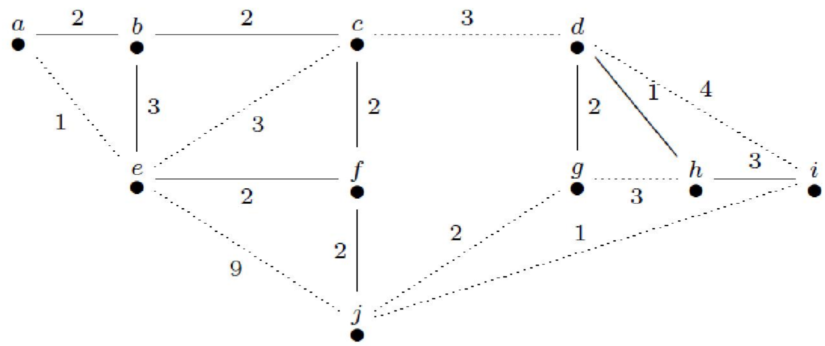


Fig. 6. The original graph *G=(V,E)*.

1. Randomly choose an edge (v_i,v_j) of G, with cost c_{ij}, as a possible edge to enter the graph;
2. Find the shortest path, d_{ij}, between v_i and v_j in G^* excluding edge (v_i,v_j) (if that is not possible, go to 1.), $d_{ij}=(v_i, w_1), (w_1, w_2), (w_2, w_3),..., (w_n,v_j)$, with cost d;

3. Create G' adding to G^* to all the edges of d_{ij} (see Fig. 7);

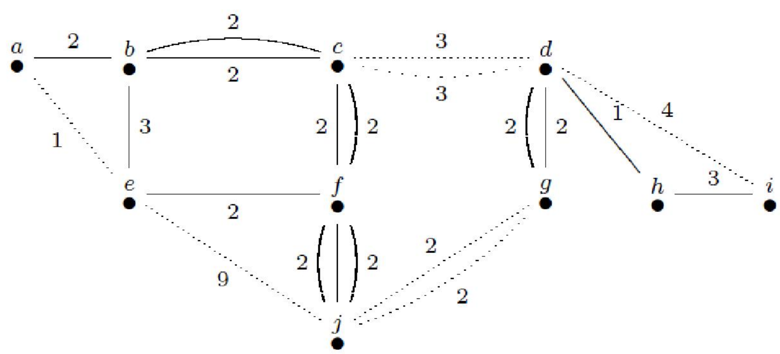


Fig. 7. *Graph $G'=G^* \cup d_{ij}$, after choosing edge (f,j).*

4. Calculate Δ, in G':

$$\Delta = c_{ij} + d - 2\times \sum_{(v_n,v_m)\in F} c_{nm} - 2\times \sum_{(v_n,v_m)\in B} c_{nm} \;;$$

(where F represents the set of facultative edges that become doubles with the introduction of d_{ij}; B is the set of edges of any type that have more than 2 copies while c_{nm} represents the cost associated to edge (v_n,v_m)).
   a. If $\Delta \geq 0$, back to *1*;
   b. If $\Delta < 0$ go to *5*;

5. Remove from G' all the facultative edges that became doubles with the introduction of d_{ij}, remove also 2 copies of edges, of any type, that have more than 2 copies;

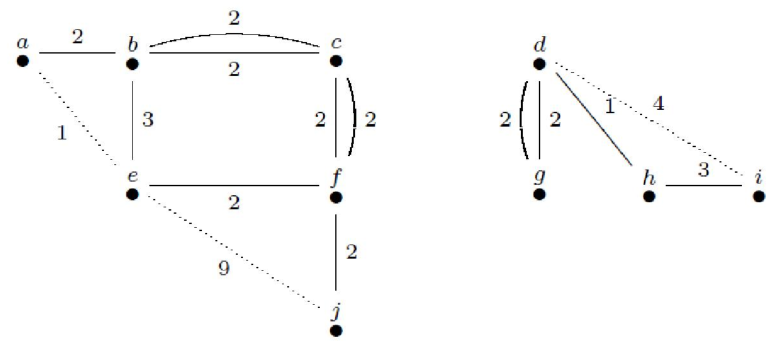


Fig. 8. New *G'* (after determination of $\Delta = -3$); *G'* is not connected.

6. Is *G'* connected (see Fig. 8)?
    a. YES, *G'* is the new solution (with $cost = y + \Delta$);
    b. NO, go to *7.*;

7. Choose and duplicate an edge e_{pq} (from the set of facultative edges of *G*) of minimum cost, c_{pq}, among those that join two connected components *p* and *q*. Thus, a new graph *G''* is obtained. Calculate $\Delta_1 = \Delta + 2 \times c_{pq}$;
    a. If $\Delta_1 \geq 0$ go to *1.* ((v_i, v_j) is not a good choice).
    b. If $\Delta_1 < 0$ then;
        i. If *G''* is not connected, go back to *7.* and do $\Delta := \Delta_1$;
        ii. If *G''* is connected, return solution (with $cost = y + \Delta_1$, see Fig. 9);

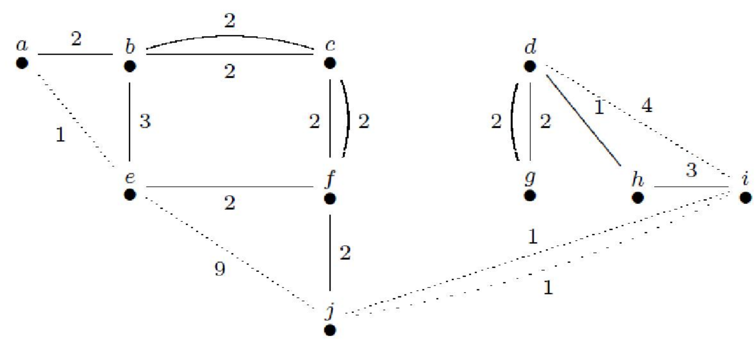


Fig. 9. *G''* is created, with $cost = y + \Delta_1 = 42 + (-3) + 2 \times 1 = 41$.

3.6 Euler's Circuit

Euler's Circuit is only created at the end of the application of MARP. *G=(V,E)* becomes an Eulerian Graph.

3.7 Pseudo Code

The pseudo code concerning the implementation of MARP:

```
BEGIN
Input (Vertices; Required Edges; Facultative Edges)
Begin Population
   Evaluate fitness of Agents
   LOCAL SEARCH
   STRUCTURE
   Repeat
      CROSSOVER, LOCAL SEARCH, STRUCTURE
      Count "No changes on a pocket"
      For i=1 to 13 (agents)
         If "No changes on a pocket [i]=k";
            Do MUTATION, LOCAL SEARCH;
         End If
      End For
      STRUCTURE
   Until stop criterion = FALSO
Output best EULER CIRCUIT
END
```

4 Computational Results

This section presents an important selection of computational experiments to evaluate MARP.

The relevance of MARP should be observed in the twofold context:

1. as a procedure to solve some industrial path cutting problems and
2. as a method to solve the RPP.

4.1 Cutting Path Applications

Four path cutting applications are illustrated below. The non required path is called the 'Empty Path' (*EP*) and the objective is to minimize the *EP*. The first three applications are illustrated in Fig. 10**.** The first three applications, Ex1, Ex2 and Ex3 are illustrated in Fig. 10 and the data (vertices, required and facultative edges and pieces) may be found at http://www.inescporto.pt/~amr/. The data include the identification of vertices, required and facultative edges and pieces (see http://www.inescporto.pt/~amr/Read me.txt).

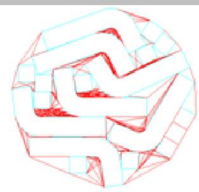
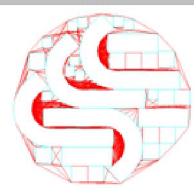
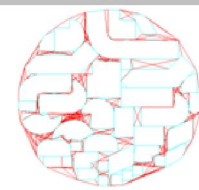
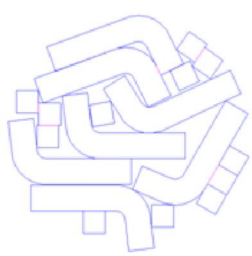
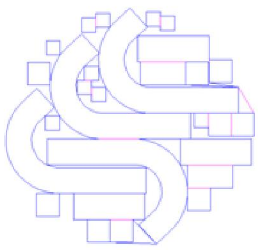
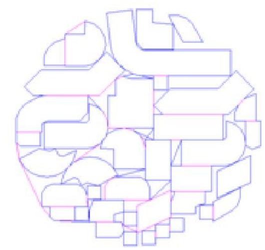
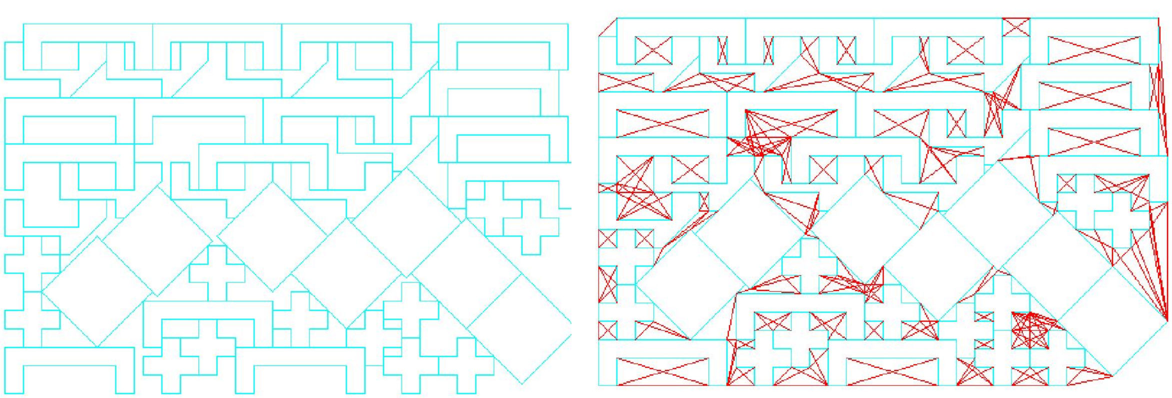
	Ex1	Ex2	Ex3
a)			
# Pieces	19	28	38
Vertices	183	302	348
Required Edges	176	281	319
b)			
Facultative Edges	281	704	374
c)			
Empty Path (%)	3	12	23

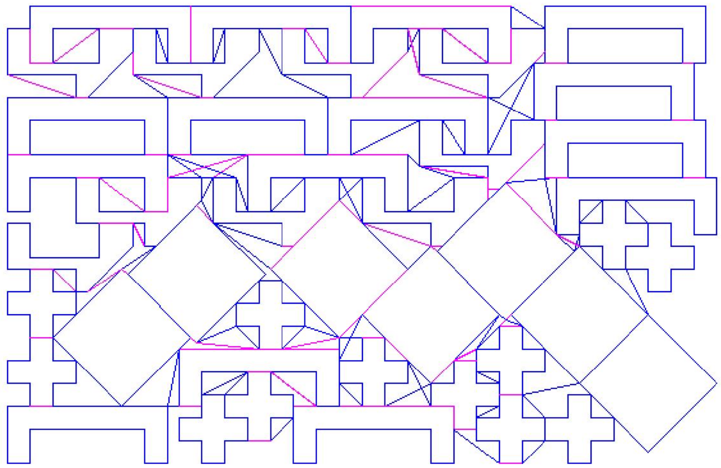
Fig. 10. (a) Nesting: Pieces to cut (required edges); **(b)** Bridges launching (non-required edges (—));
(c) Solution (Eulerian graph with simple (—) and double (—) edges).

The last application (348 vertices, 439 required edges and 286 facultative edges), presented in Fig. 11, is based on a problem described at http://paginas.fe.up.pt/~esicup/tiki-nesting_layout.php?sol=1&file=shapes0.xml [13]. The data relating to the solution with MARP may be found at http://www.inescporto.pt/~amr/Topos.txt.



a) Required edges

b) Required (—) and facultative edges (—)



(c) Solution: *EP = 31%* (Eulerian graph with simple (—) and double (—) edges).
Fig. 11. Application with 348 vertices, 439 required edges and 286 facultative edges

4.2 RPP solutions

The computational experiments and results reported are based on a series of instances that cover the most common examples in the literature and in particular the most cited ones. The aim is to prove that MARP could be a good choice when solving the RPP, due to its simplicity and the relative quality of the solutions. The results obtained are compared with different authors' methods/implementations.

They are divided into 2 sets

1. Instances of small size , including Group 1 (*P_i*) and Group 2 (*Albaida_x*), presented in Table 1 and
2. Instances of medium/big size, including Group 3 (*GRP_i*), Group 4 (*Alba_i_j*) and Group 5 (*Madr_i_j*), presented in Table 2.

Each table has 10 columns. The first column identifies the problem while the other columns refer to:

- *# Comp*: number of components connected in the graph, only considering the required edges;
- *# Vert*: number of vertices of the initial graph;
- *# RE*: number of required edges;
- *# NE*: number of non-required edges;
- $\sum RE$: the sum of the costs for all of the required edges;
- OV^*: the optimal value;
- *MARP*: the result obtained using the MARP algorithm;
- OV^*/*MARP*: ratio between the optimal value and the result of MARP;
- *Time (sec)*: computational time in seconds.

Table 1. Characteristics and results of the 24 instances P_i and the 2 instances Albaida_x (Group 1 and Group 2)

	#Comp	#Vert	#RE	#NE	$\sum RE$	OV^*	*MARP*	OV^*/*MARP*	*Time(sec)*
P01	4	11	7	6	25	76	76	1,00	<1
P02	4	14	12	21	80	152	152	1,00	8
P03	4	28	26	31	73	102	102	1,00	10
P04	3	17	22	13	55	84	84	1,00	4
P05	5	20	16	19	69	124	124	1,00	5
P06	7	24	20	26	70	102	124	1,00	10
P07	3	23	24	23	93	130	130	1,00	6
P08	2	17	24	16	93	122	122	1,00	5
P09	3	14	14	12	57	83	83	1,00	3
P10	4	12	10	10	45	80	80	1,00	10
P11	3	9	7	7	14	23	23	1,00	5
P12	3	7	5	13	13	19	19	1,00	2
P13	3	7	4	6	12	35	35	1,00	1
P14	6	28	31	48	145	202	202	1,00	15
P15	8	26	19	18	180	441	441	1,00	25
P16	7	31	34	60	139	203	203	1,00	28
P17	5	19	17	27	63	112	112	1,00	16
P18	8	23	16	21	61	146	146	1,00	23
P19	7	33	29	25	141	257	257	1,00	32
P20	7	50	63	35	282	398	398	1,00	70
P21	6	49	67	43	288	366	366	1,00	62
P22	6	50	74	110	499	621	621	1,00	60
P23	6	50	78	80	380	475	475	1,00	62
P24	7	41	55	70	292	405	405	1,00	76
Albaida_a	10	102	99	61	7295	10599	11189	0,95	145
Albaida_b	11	90	88	56	5803	8629	8853	0,98	100

The next three groups of instances have more vertices and edges. Fernández et al. [23] present a set of heuristics that show a good practical performance with these instances. They are the result of an interesting combination of ideas and developments from different authors and incorporating, in particular, tight bounds. Lower bounds are obtained when a linear programming relaxation is solved, based on a new formulation presented in Ghiani and Laporte [22]; in a second phase, an upper bound is created based on a heuristic with three steps, this is a variation of the well-known technique of Frederickson [24].

The results obtained with MARP are compared with optimal solutions, whenever possible. Otherwise, upper bound solutions are used.

In terms of the quality of the solutions, MARP succeeded in obtaining optimal solutions every time for all of the instances of Group 1 and showed very good results for Group 2. Less positive results were obtained in Groups 3 and 4 and the worst results were shown in Group 5.

It appears that MARP works better for problems with fewer connected components; this might be the reason for the results for Group 5, the group with the higher number of connected components. This may be observed when the 3 sub-groups *Madr_x_y*, for $x = 3; 5; 7$, each one with 5 instances ($y = 1; 2; \ldots; 5$) are considered: they have the same number of vertices and edges (required plus facultative) but a very different numbers of connected components. The average number of connected components of sub-groups *Madr_3_y*, *Madr_5_y* and *Madr_7_y* are 38, 23 and 4, respectively. This conclusion was expected and is supported by Orloff [48], who considered that the complexity of this type of problem depends, not only on the number of odd vertices and required vertices, but more importantly on the number of disconnected components.

The fact that MARP works better for instances with fewer connected components may be an advantage for the application described in Section 2. The algorithm is also being evaluated in this context. Indeed in that cutting application, the problems only have one connected component, since the pieces that will be cut are joined in order to minimize the waste.

Finally, and as a conclusion of the computational experiments, it must be noted that MARP may be used to handle the Industrial Application portrayed in the paper. Furthermore, a procedure based on MARP constitutes a new and successful method based on MA, that can deal with the RPP.

Table 2. Characteristics and results for the 10 instances GRP_i, 15 instances ALBA_x_y and 15 instances MADR_x_y (Groups 3,4 and 5). A bold value represents an upper bound

	#Comp	#Vert	#RE	#NE	$\sum RE$	OV^*	*MARP*	OV^*/*MARP*	*Time(sec)*
GRP1	34	116	61	113	4034	8248	9240	0,89	120
GRP 2	30	116	64	110	4708	8592	9154	0,94	120
GRP 3	34	116	61	113	3848	8047	9066	0,89	120
GRP 4	17	116	88	86	6234	8951	9220	0,98	120
GRP 5	21	116	72	102	5299	8583	9451	0,91	120
GRP 6	4	116	126	48	8893	11595	11967	0,97	120
GRP 7	26	116	52	122	3367	7049	7667	0,92	120
GRP 8	20	116	81	93	2860	7176	7734	0,93	120
GRP 9	8	116	59	115	7602	9894	10890	0,91	120
GRP 10	17	116	87	87	6138	**8849**	9126	0,97	120
ALBA-3-1	22	116	51	123	4098	7640	7738	0,99	109
ALBA-3-2	23	116	46	128	3096	**6706**	7013	0,96	110
ALBA-3-3	15	116	44	130	3899	7475	7627	0,98	105
ALBA-3-4	21	116	49	125	3632	**7276**	7496	0,97	110
ALBA-3-5	19	116	57	117	4177	7490	8184	0,92	110
ALBA-5-1	18	116	88	86	6796	11085	11414	0,97	110
ALBA-5-2	14	116	92	82	6940	10760	11232	0,96	106
ALBA-5-3	11	116	92	82	6175	9301	9549	0,97	105
ALBA-5-4	8	116	88	86	6015	9002	9275	0,97	100
ALBA-5-5	16	116	91	83	6579	9775	10535	0,93	105
ALBA-7-1	6	116	118	56	8240	11521	12034	0,96	100
ALBA-7-2	2	116	122	52	8567	11147	11577	0,96	100
ALBA-7-3	9	116	113	61	8492	11731	12230	0,96	100
ALBA-7-4	4	116	119	55	8635	11761	12315	0,96	100
ALBA-7-5	7	116	116	58	7981	11414	11715	0,97	100
MADR-3-1	42	196	86	230	6965	**11871**	16420	0,72	220
MADR-3-2	34	196	108	208	7870	**13310**	16980	0,78	220
MADR-3-3	36	196	102	214	7465	**12803**	16640	0,77	220
MADR-3-4	39	196	101	215	7695	13090	17745	0,74	220
MADR-3-5	38	196	95	221	7145	**12073**	16930	0,71	220
MADR-5-1	21	196	163	153	11320	15470	18575	0,83	200
MADR-5-2	25	196	156	160	11935	16865	20365	0,83	200
MADR-5-3	22	196	148	168	11030	15200	18445	0,82	200
MADR-5-4	23	196	152	164	11660	16300	20170	0,81	200
MADR-5-5	26	196	147	169	10755	15515	19410	0,80	200
MADR-7-1	7	196	211	105	15805	20460	23460	0,88	180
MADR-7-2	2	196	238	78	17605	22220	25040	0,89	180
MADR-7-3	6	196	219	97	16280	24105	24105	0,86	180
MADR-7-4	3	196	225	91	17000	22280	22280	0,91	180
MADR-7-5	3	196	223	93	16625	21150	23360	0,91	180

5 Conclusions

The Rural Postman Problem (RPP) is a particular Arc Routing Problem which consists of finding the minimum circuit on a graph so that a given set of required edges is traversed. It is an NP-hard problem. The paper's main contribution is a new method that solves the RPP - the MARP algorithm - based on Memetic Algorithms (MA), in connection with a new Industrial Application. As far as the authors are aware, this is the first time MA have been used to solve the RPP and path cutting applications. MA can be interpreted as a cooperative-competitive strategy to optimize agents, with the unique characteristic of attempting to integrate social knowledge using a local search procedure and, therefore, complementing the information transmitted through crossover operations. MARP, based on a ternary structure of the population, was described in detail, after introducing the necessary graph transformations. The industrial application, linked with path cutting optimization in the manufacturing of expensive woodworking tools, which was modeled as an RPP, was portrayed. The computational experiments included many instances that were tested and compared with other known approaches. According to the results and the evaluation that was conducted, it may be said that it behaved very well in general, except for some specific instances, which had relatively more connected components. However, this is not a drawback for the cutting path applications, because the graphs only have one connected component. The MARP algorithm can easily be understood and it constitutes a new method, based on MA that can deal with the RPP. Furthermore, a procedure based on MARP may be used to handle the industrial application presented in the paper.

Acknowledgements

Finantial support by *FCT – Fundação para a Ciência e a Tecnologia (Project PTDC/GES/73801/2006)*

References

1. Ghiani, G., Guerriero, G., Improta, G., Musmanno, R.: Waste collection in southern Italy: Solution of a real-life arc routing problem. International Transactions in Operational Research, 12, 135-144 (2005)
2. Ghiani, G., Laganà, D., Musmanno, R.: A constructive heuristic for the undirected rural postman problem. Computers & Operations Research, 33, 3450-3457 (2006)
3. Christofides, N., Campos, V., Corberán, A., Mota, E.: An algorithm for the rural postman problem on a direct graph. Mathematical Programming Study, 26, 155-166 (1986)
4. Eiselt, H. A., Gendreau, M., Laporte, G.: Arc routing problems (Part II): The rural postman problem. Operations Research, 43 (3), 399-414 (1995)
5. Lenstra, J., Rinnooy Kan, A. H. G.: On general routing problems. Networks, 6, 273-280 (1976)
6. Frederickson, G. N., Hecht, M. S., Kim, C. E.: Approximation algorithms for some routing problems. SIAM J Comput., 7, 178-193 (1978)
7. Manber, U., Israni,S.: Pierce point minimization and optimal torch determination in flame cutting. Journal of Manufacturing Systems, 3 (1), 81-89 (1984)
8. Ghiani, G., Improta, G.: The laser-plotter beam routing problem. Journal of the Operational Research Society, 52, 945-951 (2001)
9. Moreira, L. M., Oliveira, J. F., Gomes, A. M., Soeiro Ferreira, J.: Heuristics for a dynamic rural postman problem. Computers & Operations Research, 34 (11), 3281-3294 (2007)
10. Imahori, S., Kushiya, M., Nakashima, T., Sugihara, K.: Generation of cutter paths for hard material in wire edm. Journal of Materials Processing Technology, 206, 543-461 (2008)
11. Huang, X., Xi, F., Li, J., Zhong, Z.: Optimal layout and path planning for flame cutting of sheets metals. International Journal of Computer Integrated Manufacturing, 22 (1), 30-41 (2009)
12. Usberti, F. L., França, P. M., França, A. L. M.: The open capacitated arc routing problem. Computers & Operations Research, 38, 1543-1555 (2011)
13. Gomes, A. M., Oliveira, J. F.: Solving irregular strip packing problems by hybridising simulated annealing and linear programming. European Journal of Operational Research, 171 (3), 811-829 (2006)
14. Oliveira, J. F., Gomes, A. M., Ferreira, J. S.: Topos a new constructive algorithm for nesting problems. OR Spectrum, 22, 263-284 (2000)
15. Burke, E., Hellier, R., Kendall, G., Whitwell, G.: A new bottom-left-fill heuristic algorithm for the two-dimensional irregular packing problem. Operations Research, 54 (3), 587-601 (2006)
16. Bennell, J. A., Oliveira, J. F.: The geometry of nesting problems: A tutorial. European Journal of Operational Research, 184 (2), 397-415 (2008)
17. Eiselt, H. A., Gendreau, M., Laporte, G.: Arc routing problems (Part I): The chinese postman problem. Operations Research, 43 (2), 231-242 (1995)
18. Laporte, G., Osman, I. H.: Routing problems: A bibliography. Annals of Operations Research, 61, 227-262 (1995)
19. Orloff, C. S.: A fundamental problem in vehicle routing. Networks, 4, 35-64 (1974)
20. Christofides, N., Campos, V., Corberán, A., Mota, E.: An algorithm for the rural postman problem. Technical Report IC.OR. 81.5, Imperial College Report (1981)

21. Corberán, A., Sanchis, J. M.: A polyhedral approach for the rural postman problem, European Journal of Operational Research, 79, 95-114 (1994)
22. Ghiani, G., Laporte, G.: Branch-and-cut algorithm for the undirect rural postman problem. Mathematical Programming, 87 (3), 467-481 (2000)
23. Fernández, E., Meza, O., Garfinkel, R., Ortega, M.: On the undirected rural postman problem: Tight bouns based on a new formulation. Operations Research, 51 (2), 281-291 (2003)
24. Frederickson, G. N.: Approximation algorithms for some postman problems. Journal of the ACM, 26 (3), 538-554 (1979)
25. Pearn, W. L., Wu, T. C.: Algorithms for the rural postman problem. Computers & Operations Research, 22 (8), 818-828 (1995)
26. Laganà, D., Laporte, G., Mari, F., Musmanno, R., Pisacane, O.: An ant colony optimization metaheuristic for the undirected rural postman problem. Les Cahiers du GERAD, G-2007-106 (2007)
27. Hertz, A., Laporte, G., Hugo, P. N.: Improvement procedures for the undirect rural postman problem. INFORMS Journal on Computing, 11 (1), 53-52 (1999)
28. Garfinkel, R. S., Webb, I. R.: On crossings, the crossing problem postman problem, and the rural postman problem. Networks, 34, 173-180 (1999)
29. Córdoba, P. F., Raffi, L. M. G., Sanchis, J. M.: A heuristic algorithm based on Monte Carlo methods for the rural postman problem. Computers & Operations Research, 25 (12), 1097-1106 (1998)
30. Corberán, A., Martí, R., Romero, A.: Heuristics for the mixed rural postman problem. Computers & Operations Research, 27, 183-203 (2000)
31. Groves, G., Roux, J. L., Vuuren, J. H. V.: Network service scheduling and routing. International Transactions in Operational Research, 11, 613-643 (2004)
32. Dror, M.: Arc Routing: Theory, Solutions and Applications. Kluwer Academic Publishers, Boston (2000)
33. Corberán, A., Prins, C.: Recent results on arc routing problems: An annotated bibliography. Networks, 56 (1), 50-69 (2010)
34. Rodrigues, A. M., Ferreira, J. S.: Rural postman and related arc routing problems. Technical Report, INESC Porto (2010)
35. Norman, M., Moscato, P.: A competitive-cooperative approach to complex combinatorial search. Technical Report 790, Caltech Concurrent Computational Program, California Institute of Technology, Pasadena, California, USA (1989)
36. Forrest, S.: Genetic algorithms. ACM Computing Surveys (CSUR), 28 (1), 77-80 (1996)
37. Baker, B. M., Ayechew, M. A.: A genetic algorithm for the vehicle routing problem. Computers & Operations Research, 30, 787-800 (2003)
38. França, P., Mendes, A., Moscato, P.: Memetic algorithms to minimize tardiness on a single machine with sequence-dependent setup times. In Proceedings of the DSI'99 – 5th International Conference of the Decision Sciences Institute, 1708-1710 (1999)
39. Buriol, L., França, P. M., Moscato, P.: A new memetic algorithm for the asymetric traveling salesman problem. Journal of Heuristics, 10, 483-506 (2004)
40. Berreta, R., Moscato, P.: The number partitioning problem: An open challenge for Evolutionary Computation. In: Corne, D., Dorigo, M. Glover, F. (eds) New Ideas in Optimization, pp., 261-278, McGraw-Hill (1999)
41. Mendes, S. M., Muller, F. M., França, P. M., Moscato, P.: Comparing Meta-Heuristics Approaches for Parallel Machine Scheduling Problems with Sequence Dependent Setup Times. In Proceedings of the 15th International Conference on CAD/CAM, Robotics & Factories of the Future (1999)
42. Berretta, R., Rodrigues, L. F.: A memetic algorithm for a multistage capacitated lot-sizing problem. International Journal of Production Economics 87, 67-81 (2004)
43. Moscato, P., Norman, M. G.: A "memetic" approach for the traveling salesman problem implementation of a computational ecology for combinatorial optimization on message-passing systems. Parallel Computing and Transputer Applications, 187-194 (1992)
44. Runggeratigul, S.: Local search and memetic algorithms for knapsack problems. The 5th Metaheuristics International Conference, 25-28 (2003)
45. Lacomme, P., Prins, C., Ramdane-Chérif, W.: Evolutionary algorithms for periodic arc routing problems. European Journal of Operational Research, 165, 535-553 (2005)
46. Cotta, C.: Una vision general de los algoritmos meméticos, Rect@, 3, 139-166 (2007)
47. Moscato, P., Cotta, C.: A modern introduction to memetic algorithm. Handbook of Metaheuristics, vol. 146, International Series in Operations Research Management Science, 141-183 (2010)
48. Orloff, C. S.: On general routing problems: Comments. Networks, 6, 281-284 (1976)