An Improved Relax-and-Fix Algorithm for the Fixed Charge Network Design Problem with User-optimal Flow

Pedro Henrique González1,2, Luidi Gelabert Simonetti1, Carlos Alberto de Jesus Martinhon1, Edercarlos Santos1 and Philippe Yves Paul Michelon2

1Institute of Computing, Fluminense Federal University, Niterói, Brazil
2Laboratoire d’Informatique d’Avignon, Université d’Avignon et des Pays de Vaucluse, Avignon, France

Keywords: Network Design Problem, Dynamic Programming, Relax-and-Fix, Bi-level Problem.

Abstract: Due to the constant development of society, increasing quantities of commodities have to be transported in large urban centers. Therefore, network planning problems arise as tools to support decision-making, aiming to meet the need of finding efficient ways to perform such transportations. This paper reviews a bi-level formulation, an one level formulation obtained by applying the complementary slackness theorem, Bellman’s optimality conditions and presents an improved Relax-and-Fix heuristic, through combining a randomized constructive algorithm with a Relax-and-Fix heuristic, so high quality solutions could be found. Besides that, our computational results are compared with the results found by an one-level formulation and other heuristics found in the literature, showing the efficiency of the proposed method.

1 INTRODUCTION

The Fixed Charge Network Design Problem (FCNDP) involves selecting a subset of edges from a graph, in such a way that a given set of commodities can be transported from their origins to their destinations. The problem consists in minimizing the sum of the fixed costs (due to selected edges) and variable costs (depending on the flow of goods on the edges). Fixed and variable costs can be represented by linear functions and arcs are not capacitated. The FCNDP belongs to a large class of network design problems (Magnanti and Wong, 1984). In the literature, one can find several variations of FCNDP (Boesch, 1976) such as shortest path problem, minimum spanning tree problem, vehicle routing problem, traveling salesman problem and network Steiner problem (Magnanti and Wong, 1984). Moreover, as illustrated by several books and papers (Boesch, 1976) (Boyce and Janson, 1980) (Mandl, 1981), generic network design problem has numerous applications. Mathematical formulations for FCNDP not only represent the FCNDP, but also problems of communication, transportation, sewage systems and resource planning. It also appears in other contexts, such as flexible production systems (Kimemia and Gershwin, 1978) and automated manufacturing systems (Graves and Lamar, 1983). Finally, network design problems arise in many vehicle fleet applications that do not involve the construction of physical facilities, but rather model decision problems such as sending a vehicle through a road or not (Simpson, 1969); (Magnanti, 1981).

In network planning problems, not only the simplest versions are NP-Hard (Johnson et al., 1978);(Wong, 1978), but also the task of finding feasible solutions (for problems with budget constraint on the fixed cost) is extremely complex (Wong, 1980). Due to the natural difficulties of the problem, heuristics methods are presented as a good alternative in the search for quality solutions.

In the paper, we intend to address a specific variation of FCNDP. The Fixed-Charge Uncapacitated Network Design Problem with User-optimal Flows (FCNDP-UOF), which consists of adding multiple shortest path problems to the original problem. The FCNDP-UOF can be modeled as a bilevel discrete linear programming problem. This type of problem involves two distinct agents acting simultaneously rather than sequentially when making decisions. On the upper level, the leader (1st agent) is in charge of choosing a subset of edges to be opened in order to minimize the sum of fixed and variable costs. In response, on the lower level, the follower (2nd agent) must choose a set of shortest paths in the network, resulting in the paths through which each commod-
ity will be sent. The effect of an agent on the other is indirect: the decision of the followers is affected by the network designed on the upper level, while the leader’s decision is affected by variable costs imposed by the routes setted in the lower level.

The inclusion of shortest path problem constraints in a mixed integer linear programming is not straightforward. Difficulties arise both in modeling and designing efficient methods. As far as we know, there are few works done on FCNDP-UOF in the literature, and most of them address to a particular variant. This problem or its variant could be seen on (Billheimer and Gray, 1973); (Kara and Verter, 2004); (Erkut et al., 2007); (Mauttone et al., 2008); (Erkut and Gzara, 2008); (Amaldi et al., 2011); (González et al., 2013) and has been treated as part of larger problems in some applications on (Holmberg and Yuan, 2004).

The FCNDP-UOF problem appears in the design of a road network for hazardous materials transportation (Kara and Verter, 2004); (Erkut et al., 2007); (Erkut and Gzara, 2008) and (Amaldi et al., 2011). During the solution of this problem the government defines a selection of road segments to be opened/closed to the transportation of hazardous materials assuming that hazmat shipments in the resulting network will be done along shortest paths. There are no costs associated with the selection of roads to compose the network but the government wants to minimize the population exposure in case of an incident during a dangerous-goods transportation. This is a particular case of the FCNDP-UOF problem where the fixed costs are equal to zero.

It is interesting to specify the contributions of each work cited above. (Billheimer and Gray, 1973) present and formally define the FCNDP-UOF. (Kara and Verter, 2004) and (Erkut et al., 2007) works focus on exact methods, presenting a mathematical formulation and several metrics for the hazardous materials transportation problem. (Mauttone et al., 2008) not only presented a different model, but also presented a Tabu Search for the FCNDP-UOF. Both, (Erkut and Gzara, 2008) and (Amaldi et al., 2011) presented heuristic approaches to tackle the hazardous materials transportation problem. At last, (González et al., 2013), presented a extension of the model proposed by Kara and Verter and also a GRASP.

This text is organized as follows. In Section 2, we start by describing the problem followed by a bi-level and an one-level formulation, presented by (Mauttone et al., 2008). Then in Section 3 we present our solution approach. Section 4 reports on our computational results. In Section 5 we will compare our results with the mathematical formulation and with heuristic results found in the literature. At last, in Section 6 the conclusion and future works are presented.

2 GENERAL DESCRIPTION OF FCNDP-UOF

In this section we describe the problem and present a bi-level and an one-level formulation for the FCNDP-UOF proposed respectively by (Colson et al., 2005) and (Mauttone et al., 2008) for the FCNDP-UOF, which we address as MLF Model. Since the structure of the problem can be easily represented by a graph, the basic structures to create a network are a set of nodes $V$ that represents the facilities and a set of uncapacitated and undirected edges $E$ representing the connection between installations. Furthermore, the set $K$ is the set of commodities to be transported over the network, and these commodities may represent physical goods as raw material for industry, hazardous material or even people. Each commodity $k \in K$, has a flow to be delivered through a shortest path between its source $o(k)$ and its destination $d(k)$. The formulation presented here works with variants presenting commodities with multiple origins and destinations, and for treating such a case, it is sufficient to consider that for each pair $(o(k), d(k))$, there is a new commodity resulting from the dissociation of one into several commodities.

2.1 Mathematical Formulation

This subsection shows a small review of FCNDP-UOF in order to exemplify the characteristics and make easier the understanding of it.

The model for FCNDP-UOF has two types of variables, one for the construction of the network and another related to representing the flow. Let $y_{ij}$ be a binary variable, we have that $y_{ij} = 1$ if the edge $(i, j)$ is chosen as part of the network and $y_{ij} = 0$ otherwise. In this case, $x_{ij}^k$ denotes the commodity $k$ flow through the arc $(i, j)$. Although the edges have no direction, they may be referred to as arcs, because each commodity flow is directed. Treating $y = (y_{ij})$ and $x = (x_{ij}^k)$, respectively, as vectors of adding edge and flow variables, a mixed integer programming formulation can be elaborated.
2.1.1 List of Symbols

\begin{itemize}
  \item $V$ Set of Nodes.
  \item $E$ Set of admissible bi-directed Edges.
  \item $K$ Set of Commodities.
  \item $\delta_i^+$ Set of all arcs leaving node $i$.
  \item $\delta_i^-$ Set of all arcs arriving at node $i$.
  \item $c_e$ Length of edge $e$.
  \item $o(k)$ Origin node for commodity $k$.
  \item $d(k)$ Destination node for commodity $k$.
  \item $\delta_{ij}^k$ Variable cost of transporting commodity $k$ through the edge $(i, j) \in E$.
  \item $f_{ij}$ Fixed cost of opening the edge $(i, j) \in E$.
  \item $y_{ij}$ Indicates if edge $(i, j)$ belongs in the solution.
  \item $x_{ij}^k$ Indicates if commodity $k$ passes through the arc $(i, j)$.
\end{itemize}

2.1.2 Bi-level Formulation

FCNNDP-UOF is a variation of the FCNDP where each $k \in K$ has to be transported through a shortest path between its origin $o(k)$ and its destination $d(k)$. This change entails adding new constraints to the general problem. In FCNNDP-UOF, besides selecting a subset of $E$ whose sum of fixed and variable costs is minimal (leading problem), each commodity $k \in K$ must be transported through the shortest path between $o(k)$ and $d(k)$ (follower problem). The FCNNDP-UOF belongs to the class of NP-Hard problems and can be modeled as a bi-level discrete integer programming problem (Colson et al., 2005), as follows:

\begin{align}
\min & \sum_{(i,j) \in E} f_{ij}y_{ij} + \sum_{k \in K} \sum_{(i,j) \in E} \delta_{ij}^k y_{ij}, \\
\text{s.t.} & \quad \sum_{(i,j) \in E} y_{ij} = 1, \\
& \quad y_{ij} \in [0, 1], \quad \forall e = (i, j) \in E,
\end{align}

where $x_{ij}^k$ is a solution of the problem:

\begin{align}
\min & \sum_{k \in K} \sum_{(i,j) \in E} c_{ij} x_{ij}^k, \\
\text{s.t.} & \quad \sum_{(i,j) \in E} x_{ij}^k - \sum_{(j,i) \in E} x_{ji}^k = \delta_{ij}^k, \quad \forall i, j \in V, \forall k \in K, \\
& \quad x_{ij}^k + y_{ij} \leq \gamma_{ij}, \quad \forall (i, j) \in E, \forall k \in K, \\
& \quad x_{ij}^k \geq 0, \quad \forall e = (i, j) \in E, \forall k \in K,
\end{align}

where:

\begin{equation}
\delta_{ij}^k = \begin{cases} 
-1 & \text{if } i = d(k), \\
1 & \text{if } i = o(k), \\
0 & \text{otherwise}.
\end{cases}
\end{equation}

Analyzing the model described by constraints (1) - (4), we can see that the set of constraints (1) ensures that $y_{ij}$ assume only binary values. In (2), we have flow constraints. Constraints (3) do not allow flow into arcs whose corresponding edges are closed. Finally, (4) imposes the non-negativity restriction of the variables $x_{ij}^k$. An interesting remark is that solving the follower problem is equivalent to solving $|K|$ shortest paths problems independently.

2.1.3 One-level Formulation

The FCNNDP-UOF can be formulated as an one-level integer programming problem replacing the objective function and the constraints defined by (2), (3) and (4) of the follower problem for its optimality conditions (Mauttone et al., 2008). This could be done by applying the fundamental theorem of duality and the complementary slackness theorem (Bazaraa et al., 2004). However, optimality conditions for the problem in the lower level are, in fact, the optimality conditions of the shortest path problem and they could be expressed in a more compact and efficient way if we consider the Bellman’s optimality conditions for the shortest path problem (Ahuja et al., 1993) and using a simple lifting process (Luigi De Giovanni, 2004). Unfortunately this new formulation loses the interesting feature of being linear. To bypass this problem a Big-M linearization is applied. After these modifications, one can write the model as an one-level mixed integer linear programming problem, as follows:

\begin{align}
\min & \sum_{(i,j) \in E} f_{ij}y_{ij} + \sum_{k \in K} \sum_{(i,j) \in E} \delta_{ij}^k y_{ij}, \\
\text{s.t.} & \quad \sum_{(i,j) \in E} x_{ij}^k - \sum_{(j,i) \in E} x_{ji}^k = \delta_{ij}^k, \quad \forall i, j \in V, \forall k \in K, \\
& \quad x_{ij}^k + y_{ij} \leq \gamma_{ij}, \quad \forall (i, j) \in E, \forall k \in K, \\
& \quad x_{ij}^k \geq 0, \quad \forall e = (i, j) \in E, \forall k \in K,
\end{align}

where:

\begin{equation}
\delta_{ij}^k = \begin{cases} 
-1 & \text{if } i = d(k), \\
1 & \text{if } i = o(k), \\
0 & \text{otherwise}.
\end{cases}
\end{equation}

The variables $\pi_i$, $k \in K$, $i \in V$, are the shortest distance between vertex $i$ and vertex $d(k)$. Then we define that $\pi_i^k$ will always be equal to zero. Assuming $y$ and $x$ binary and assuming that the inequalities (7) are satisfied, it is easy to see that constraints (8) are equivalent to Bellman’s optimality conditions for a $|K|$ set of pairs $(o(k), d(k))$. 

102
3 SOLUTION APPROACH

We address this section to present and explain the Partial Decoupling Heuristic and the Relax and Fix Heuristics. Before explaining the improved Relax-and-Fix heuristic, called DPRF, a small review of the Relax-and-Fix heuristic is presented.

3.1 Partial Decoupling Heuristic

A total decoupling heuristic for the FCNDP-UOF is based on the idea of dissociating the problem of building a network from the shortest path problem. However, as discussed in (Erkut and Gzara, 2008), the decoupling of the original problem can provide worst results than when addressing both problems simultaneously. Therefore, this algorithm proposes what we call partial decoupling, where certain aspects of the follower problem are considered when trying to build a solution to the leading problem. So in order to build the network the following cost is used:\( (\beta_{ij}^k + (1 - \alpha) \times e) \), which means that we consider whether the is edge open or not, plus a linear combination of the variable cost and the length of the edge. The \( \alpha \) works as a mediator of the importance of the \( g_{ij}^k \) and \( c_e \) values. In the beginning of the iterations \( \alpha \) prioritizes the variable cost \( (g_{ij}^k) \), while in the end it prioritizes the edge length \( (c_e) \). After building the network, a shortest path algorithm is applied to take every product from its origin to its destination \( d(k) \), considering \( c_e \) as the edge cost. It is important to note that \( g_{ij}^k = q_e^k \beta_{ij} \), where \( q_e^k \) represents the amount of commodity \( k \) and \( \beta_{ij} \) represents the shipping cost through the edge \( e = (i, j) \).

The algorithm presented here is a small variation of the Partial Decoupling Heuristic presented in (González et al., 2013). The procedure is further explained on Algorithm 1.

The partial decoupling heuristic consists in using the Dijkstra algorithm for the shortest path problem. Procedures DijkstraLeader and DijkstraFollower sequentially solve the problem of network construction, followed by the shortest path problem for each commodity \( k \in K \), so that in the end of the procedure, all commodities have been transported from its origin to its destination. The DijkstraLeader and DijkstraFollower procedures costs. The notation \( s \leftarrow y, x \rightarrow \) represents that the solution \( s \) is storing the values of the variables \( y \) and \( x \) that were just defined by DijkstraLeader and DijkstraFollower. The function CloseEdge closes all the edges that at the end of the DijkstraFollower procedure are open and do not have flow. The random function returns a random element from the set passed as a parameter. In order to choose the insertion order of \( |K| \) commodities, the procedure uses a candidate list consisting of a subset of products not yet routed, whose amount is greater than or equal to \( \gamma \) times the largest amount of commodity not routed. The function Rearm \((K)\) adds all commodities to set \( K \) and makes all variables return to its initial state.

3.2 Relax and Fix Heuristic

Given a mixed integer programming formulation:

\[
\begin{align*}
\min \quad & c_1^1 z_1^1 + c_2^1 z_2^1; \\
\text{s.t.} \quad & A^1 z_1^1 + A^2 z_2^1 = b; \\
& z_1^1 \in \mathbb{Z}_{n_1}^N, z_2^1 \in \mathbb{Z}_{n_2}^N;
\end{align*}
\]

(14)

(15)

without loss of generality, let’s suppose that the variables \( z_1^j \) for \( j \in N_1 \) are more important than the variables \( z_2^j \) for \( j \in N_2 \), with \( n_i = |N_i| \) for \( i = 1, 2 \).

The idea of the Relax and Fix, consists in solving two (or more) easier LPs or MIPs. The first one allows us to fix (i.e. \( z_1^j = w, w \in \mathbb{Z}_{n_1}^N \)) or limit the range of more important variables, while the second allows us to choose good values for other variables \( z_2^j \).

In order to do so, first it is necessary to solve a relaxation like:

```
Algorithm 1: Partial Decoupling Heuristic.

Input: \( \gamma \)
Data: \( \text{MinCost} \leftarrow \infty, \alpha \leftarrow 1, y \leftarrow 0, x \leftarrow 0; \)
begin
\( K \leftarrow K; \)
for \( \text{numIterDP} \) in \( 1 \ldots \text{MaxIterDP} \) do
while \( K \neq \emptyset \) do
\( K' \leftarrow \text{CandidateList}(K, \gamma); \)
\( k' \leftarrow \text{Random}(K'); \)
for each \( e = (i, j) \in E \) do
\( \text{DLCost}(e, k') \leftarrow (\beta_{ij}^k + (1 - \alpha) \times e); \)
\( y \leftarrow \text{DijkstraLeader}(\text{DLCost}, k'); \)
\( K \leftarrow K' \setminus \{k'\}; \)
for each \( e = (i, j) \in E \) do
\( \text{DSCost}(e) \leftarrow c_e; \)
\( x \leftarrow \text{DijkstraFollower}(\text{DSCost}, k'); \)
CloseEdge(x);
if Cost(s) < MinCost then
\( s_{\text{best}} \leftarrow s; \)
\( \text{MinCost} \leftarrow \text{Cost}(s_{\text{best}}); \)
\( \alpha \leftarrow \alpha - \frac{1}{\text{MaxIterDP}}; \)
\( \text{Rearm}(K); \)
return \( s_{\text{best}} \)
end
```

we followed the order in which the commodities appeared in the instance. The function \( \text{SolveLR}(V, E, K, \text{MinCost}) \) solves the linear relaxation of the MLF Model for the sets \( V, E \) and \( K \), taking into consider the primal bound \( \text{MinCost} \). The \( \text{RCVF}(\gamma_e) \) function returns \( \text{TRUE} \) if the Linear Relaxation cost plus the Reduced Cost of \( \gamma_e \) is lower than the current Relax and Fix solution. Since \( \gamma_e \) and \( x^e \) are decision variables in the integer programming model, the function \( \text{Feas}(s) \) returns true if the solution \( s \) passed as parameter is a feasible solution to the problem and returns false otherwise.

### Algorithm 3: DPRF.

Data: \( \text{MinCost} \leftarrow \infty \)

begin
\[ s \leftarrow \text{PartialDecoupling}(\gamma); \]
\[ \text{MinCost} \leftarrow \text{Cost}(s); \]
for \( k \in K \) do
\[ \text{for } e \in E \text{ do} \]
\[ x^e \leftarrow \text{SolveLR}(V, E, K, \text{MinCost}); \]
\[ \text{for } e \in E \text{ do} \]
\[ \text{if } \gamma_e = 0 \text{ and } \text{RCVF}(\gamma_e) = \text{TRUE} \text{ then} \]
\[ y_e \leftarrow 0; \]
\[ \text{if } \text{Cost}(s) < \text{MinCost and Feas}(s) = \text{TRUE} \text{ then} \]
\[ s_{\text{best}} \leftarrow s; \]
\[ \text{MinCost} \leftarrow \text{Cost}(s_{\text{best}}); \]
return \( s_{\text{best}} \)

Since the Partial Decoupling Heuristic provides a feasible solution, no recovery strategy was developed in case the current fixing of the variables turns out to be infeasible.

### 4 COMPUTATIONAL RESULTS

In this section we present computational results for the one-level model and for the Relax-and-Fix presented in the previous section. The algorithms were coded in Xpress Mosel using FICO Xpress Optimization Suite, on an Intel (R) Core TM 2 CPU 6400@2.13GHz computer with 2GB of RAM. Computing times are reported in seconds. In order to test not only the performance of the one-level model, but also the performance of the presented heuristic, we used networks data obtained through private communication with one of the authors of
| Table 1: Computational results for Tabu Search and GRASP approach. |
| Opt | Tabu Search MFL | GRASP |
| Best | Best | Time | GAP | Avg | Sol | Time | GAP | Avg | Sol | Time | GAP | Avg | Sol | Time | GAP |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 30-0-8-30-001 | 6989 | 7522 | 93 | 0.048 | 7122.2 | 328.295 | 182.39 | 4.115 | 6989 | 325.357 | 0.000 |
| 30-0-8-30-002 | 7746 | 8142 | 565 | 0.051 | 8124 | 337.191 | 16.43 | 33.634 | 8112 | 321.838 | 0.047 |
| 30-0-8-30-003 | 8284 | 8854 | 1287 | 0.055 | 8384 | 318.062 | 0 | 26.991 | 8384 | 338.249 | 0.000 |
| 30-0-8-30-004 | 7428 | 7502 | 794 | 0.010 | 7442.8 | 321.434 | 33.09 | 17.889 | 7428 | 344.367 | 0.000 |
| Avg | 769.8 | 0.04 | 327.42 | 0.01 |

| Table 2: Computational results for Tabu Search and DPRF approach. |
| Opt | Tabu Search MFL | DPRF |
| Best | Best | Time | GAP | Avg | Sol | Time | GAP | Avg | Sol | Time | GAP | Avg | Sol | Time | GAP |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 30-0-8-30-001 | 6989 | 7322 | 93 | 0.048 | 7322 | 338.52 | 0 | 26.991 | 7322 | 338.249 | 0.000 |
| 30-0-8-30-002 | 7746 | 8142 | 565 | 0.051 | 8112 | 337.191 | 16.43 | 33.634 | 8112 | 321.838 | 0.047 |
| 30-0-8-30-003 | 8284 | 8854 | 1287 | 0.055 | 8384 | 318.062 | 0 | 26.991 | 8384 | 338.249 | 0.000 |
| 30-0-8-30-004 | 7428 | 7502 | 794 | 0.010 | 7442.8 | 321.434 | 33.09 | 17.889 | 7428 | 344.367 | 0.000 |
| Avg | 769.8 | 0.04 | 327.42 | 0.01 |

| Table 3: Computational results for GRASP and DPRF approach. |
| Opt | GRASP | DPRF |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Avg | 769.8 | 0.04 | 327.42 | 0.01 |
The instances are grouped according to the number of nodes in the graph (10, 20, 30), followed by the graph density (0.3, 0.5, 0.8) and finally the amount of different commodities to be transported. For the presented tables, we report the optimum value found by exact model (Opt), the best solution (Best Sol) and best time (Best Time) reached by selected approach, and the gap value between exact and heuristic (GAP). We also reported the average values for time (Avg Time) and for solutions (Avg Sol). Finally, reported standard deviation values for time (Dev Time) and solution (Dev Sol). In both tables the results in bold represent the best solution found, while the underlined ones represent that the optimum has been found.

In Table 1 and 2, we present the results reached for the instances generated by (Mauttone et al., 2008). For these five instances, three heuristics were compared: the Tabu Search heuristic proposed by (Mauttone et al., 2008), the GRASP heuristic of (González et al., 2013) and the DPRF algorithm. For the Tabu Search, the average time was high and no optimum solution was found. When observing the gap value, the table shows that the GRASP heuristic obtained best solutions in general, however the computational time is very high in comparison with the DPRF heuristic. Moreover, the standard deviation obtained by GRASP presented high values suggesting the algorithm has an irregular behavior and for the DPRF algorithm all standard deviation values for solutions were 0. Although for those instances GRASP outperform the DPRF in solution quality (3 out of 5), table 2 shows that DPRF outperform the Tabu Search presented by (Mauttone et al., 2008).

In Table 3 were used another 45 instances generated by Mauttone, Labb and Figueiredo, whose results were not published by them. For this group of instances, the computational results suggest the efficiency of DPRF heuristic. On average, the DPRF was 20 times faster than GRASP. Also, DPRF found 29 optimal solutions, while GRASP found only 7 optimal solutions. Besides that, the DPRF also improved or equaled GRASP results for 40 (36 improvements) out of 45 instances.

5 CONCLUSIONS AND FUTURE WORKS

We proposed a new algorithm for a variant of the fixed-charge uncapacitated network design problem where multiple shortest path problems are added to the original problem. In the first phase of the algorithm, the Partial Decoupling heuristic is used to build an initial solution. In the second phase, a Relax and Fix heuristic is applied to improve the solution cost. The proposed approach was tested on a set of instances grouped by graph density, number of nodes and commodities. Our results have shown the efficiency of DPRF in comparison with a GRASP and Tabu Search heuristic, once that the proposed algorithm presented best average time for all instances, often reaching optimum solutions. In a few cases, GRASP reached best solution values, however the computational time spend was not good when compared with DPRF.

As future work, we intend to work on exact approaches as Benders decomposition and Lagrangian relaxation since both are very effective for similar problems, as could be seen in (Bektas et al., 2007) and (Costa et al., 2007).

ACKNOWLEDGEMENTS

This work was supported by CAPES (Process Number: BEX 9877/13-4) and by Laboratoire d’Informatique d’Avignon, Université d’Avignon et des Pays de Vaucluse, Avignon, France.

REFERENCES


Costa, A. M., Cordeau, J.-F., and Gendron, B. (2007). Benders, metric and cutset inequalities for multicommodi-


