THE PUBLISHING HOUSE
OF THE ROMANIAN ACADEMY

PROCEEDINGS OF THE ROMANIAN ACADEMY, Series A,
Volume 10, Number 1/2009, pp. 000–000

ROUTING ELASTIC DEMANDS: A POLYMORPHISM BASED METAHEURISTIC SCHEME

Alain QUILLIOT , Loïc YON

Université Blaise Pascal - LIMOS/ISIMA - Campus des Cézeaux BP10125 - 63173 Aubière CEDEX, France
Corresponding author: alain.quilliot@isima.fr

We present here a Demand Maximizing Circuit Problem, which involves time elastic demands, and which is related to applications of Network Synthesis to the design of urban public transportation systems. This problem consists in the optimization, on some irregular domain, of some quantity whose computation involves heavy computational costs. We propose a specific metaheuristic Pursuit scheme, based upon the application to the original problem of a multiform rewriting process. We present and discuss various interpretations of this scheme together with practical experimentations. The whole paper is an extension of a work which was presented during the LT '2007 conference in Sousse.

Key words: Routing, elasticity, metaheuristic.

1. INTRODUCTION

We address here a specific transportation problem which we call *Demand Maximizing Circuit* problem (**DMC**). This problem, which we are going to study through the prism of software design, must be viewed as part of a more general transportation network design problem. We consider a network, which is likely to represent some urban transit infrastructure network, and we want to design the routes which are going to be run by a flexible shuttle fleet, in response to a demand which is supposed to depend on the quality of the induced connections. According to our model, those routes are going to be represented through a circuit family Γ, given together with some kind of schedule. Then we assume that the running cost of this fleet essentially depend on the lengths of the circuits γ of Γ. Also, we suppose that the global access demand *Global-Demand*(Γ), which is generated by this shuttle fleet, may be expressed as a sum $\Sigma_{i\in I} D_i(\Gamma)$ of local demands $D_i(\Gamma)$ related to a large set of origin/destination node pairs (o_i,d_i), $i \in I$, and that every local demand $D_i(\Gamma)$ is *time elastic*, which means that it depends on the time $T_i(\Gamma)$ that a user of this shuttle service will need in order to go from o_i to d_i. Of course, our goal is here to design Γ in such a way that it maximizes the global demand *Global-Demand*(Γ), while keeping running costs under control. Getting software tools for this general network synthesis problem might help planners in getting an evaluation of the opportunity of the application of reorganization scenario to a flexible public transportation system.

The main specificity of this problem is related to the time dependency of the access demand which is induced by our system. While many authors have already worked on route design problems related to flexible public transportation systems (see for instance [1-9], for the *Pick-up-and-delivery problem*, *Dial-a-ride problem*, *One period-bus-touring problem*…), few of them have taken into account demand variability, partly because getting models for this variability is something difficult (see for instance [10-13]).

As a matter of fact, the above general network design problem may be handled, in a heuristic way, through a decomposition scheme, which give rise to an eventually large sequence of calls to a process which focuses on the specification of a single circuit γ of set Γ, while considering the rest of the system as being fixed. So, we are going to devote this paper, which is an extension of a previous work (see [14]) which was presented in 2007 during the LT'2007 conference in Sousse, Tunisia, not to the general problem, but to a thorough study of this specific circuit search sub-problem, which will be called **DMC:** *Demand Maximizing*

Circuit problem. The fact is that **DMC** problem, though it is simple to set, gives rise to several interesting questions about programs and software. First, its input data (costs, demands…), as well as the model itself, cannot be anything else than an approximation, while instances of **DMC** may eventually be tackled a large number of times throughout a process which aims at dealing with the general network design problem. That means that computing acceptable solution in a short time will be more important for us than spending a lot of time in getting high precision solutions. Also, we will check that the structure of the search domain of **DMC** may very irregular, compelling the programmer to cope with a large number of local optima, most of them inducing poor solutions. But the most important point is that getting an evaluation of the quality criterion of **DMC** (essentially the induced demand) requires the resolution of a large amount of shortest path sub-problems and thus induces high computational costs. Consequently, *stricto sensu* applying to the **DMC** problem a probabilistic rewriting scheme as the *Simulated Annealing* scheme (see for instance [15-16]), or an other kind of classical local search heuristic scheme (see for instance [17-18]), which essentially consists in the introduction of some random noise every time a quantity *Global-Demand*(γ) is computed, does not really fit our problem. Instead, we are going to propose a *rewriting* approach which will aim at reducing computational costs while regularizing the structure of the domain search (Look for an analogy with, for instance [19-22]): this approach will be based upon the introduction of a family of auxiliary quality criteria, which will involve low computational costs and which will simultaneously work as approximations and pseudo-random perturbations of the original quality *Global-Demand*.

So the paper will be organized as follows: Section 2 and 3 will be devoted to a description of the DMC model, of the basic algorithmic tools which we need in order to deal with it, and to a short description of standard local search and branch and bound scheme which we tested on it. Next, Section 4 will allow us to describe the pseudo-random approximation rewriting scheme which we designed in order to handle the above mentioned difficulties, and final Section 5 will describe the experimentation process which we performed in order to test this new approach.

2. THE DEMAND MAXIMIZING CIRCUIT PROBLEM (DMC)

As it was told during the introduction, we were led to work, in the context of a partnership between our laboratory and the **CERTU** (*Centre d'Etudes sur les Réseaux de Transports Publics*) on mathematical and computing tools which could eventually help planners in proposing reorganization scenario for public flexible transportation services (services based upon the use of shuttle fleets, micro-vehicles, semi-autonomous vehicles…). In order to do it, we used a representation of the transit infrastructure as a network G = (X, E), whose arcs were labeled with technical characteristics: transportation modes, congestion rates, capacities, expected running times…). Then, a flexible transportation service was supposed to be determined, prices being fixed, by a set of routes and related schedules. Since the study was partly motivated by considerations related to economics, we had to introduce, as an **essential component of our model**, the dependency of the access demands to the quality of the service. So we supposed that this access demand was essentially dependent on the transit times induced by the service for the users between the nodes of the urban network, and that it could be expressed through some family OD of origin-destination pairs of vertices, given together with some function K allowing us to compute, for any pair (x, y) in OD, the expected transportation demand related to a move from x to y whose expected duration would be equal to some value t(x,y). By doing this, we could set a general **Strategic Transportation Network Design** problem **(STNDP)**, where the unknown object was a collection of routes, i.e. of triples (*Circuit, Vehicle Class, Frequency*), subject to several *Quality of Service* constraints and required to optimize some compound criterion, involving both running costs and global induced demand.

Handling this complex large scale problem had to be done while using decomposition techniques: we used a decomposition scheme which made appear, at the core of the **STNDP** problem, some "atomic" ***Demand Maximizing Circuit*** (**DMC**) sub-problem involving the search of a single circuit, submitted to a threshold constraint related to its length and required to maximize some *additional induced demand*. This specific **DMC** sub-problem could be viewed as one of the two elementary components of our decomposition scheme, the other one consisting in scheduling and assigning the vehicles on a given circuit family. The general decomposition scheme which we used in order to deal with the **STNDP** problem was such that the

DMC component had to be handled several times, eventually a large number of times, throughout the execution of the general resolution process. This **DMC** problem, which is going to be now at the center of this paper, can be formulated as follows:

The Demand Maximizing Circuit Problem DMC(k)

When we deal with this problem, we suppose that we are provided with the following input data:

Input Data for the DMC(k) Problem

- A network G = (X,E), which is supposed to represent the current transit infrastructure: thus, solving **DMC(k)** will aim at constructing a new *shuttle route* on G;
- A *length threshold* parameter k: the length of a new route is not going to exceed the value k;
- a function *Shuttle-Time* T_{NA} which, to any arc e in E, makes correspond some expected duration $T_{NA}(e)$ of the transit of a vehicle of the new route along the arc e;
- a function *No-New-Time* T_P which, to any vertex pair (x,y) in X.X, makes correspond some expected time $T_P(x,y)$ for a user which would go from x to y without using the new route: we suppose that, for any arc e = [x, y], we have $T_P(x,y) \le T_{NA}([x, y])$;
- a *standard frequency* w_0 : vehicles running along a new route are likely to do it according to a mean frequency equal to w_0. Thus, the mean induced waiting time will not exceed 1/2 w_0.
- a (large scale) set OD = $\{(x_i,y_i), i \in I\}$, of origin/destination pairs of vertices of G;
- related *Maximal Demand* coefficients D^*_i, $i \in I$: coefficient D^*_i, $i \in I$ is going to provide us with an evaluation of the maximal number of users which might ask for an access to the new shuttle route, whatever be the way this route is designed;
- an *Elasticity Function* K, from $\Delta = \{(u, v) \text{ such that } u \le v\} \subset R^2$ to [0, 1], such that:
  - K(u, v) decreases from 1 to 0 when v grows from u to $+\infty$;
  - K(u, v) converges to 0 when u converges to 0, (while v remains fixed).

  The meaning of this function K is that if $t^\gamma(x_i,y_i)$ denotes the expected time which is required from a user of the shuttle service when going from origin x_i to destination y_i while using the new route γ, and if $T_{NA}^*(x_i,y_i)$ denotes the Shuttle T_{NA}–shortest-path distance from x to y, then only $D_i^*.K(T_{NA}^*(x_i,y_i), t^\gamma(x_i,y_i))$ users are going to ask for an access to the new route γ when trying to go from x_i to y_i .

Related additional notations.

If x and y are two vertices of G, we denote by $T_{NA}^*(x,y)$ the shortest path distance from x to y which is defined by the T_{NA} *Shuttle-Time* function. A related shortest path will be called a *Geodesic Path*.

If γ is a circuit of G, and if x and y are two vertices of γ, then following γ from x to y means travelling along a path $\gamma_{x,y}$ which will be called the *restriction* of γ from x to y. The T_{NA}-length of this path is then denoted by T_{NA}-Length($\gamma_{x,y}$).

- **Output object for DMC(k) Problem :** according to these hypothesises, we want to compute some circuit γ of G whose T_{NA}-length does not exceed the value k, and which is going to maximize the following quantity *Global-Demand*(γ), which is defined by :

$$\textit{Global-Demand}(\gamma) = \sum_{i \in I} D_i^*.K(T_{NA}^*(x_i,y_i), t^\gamma(x_i,y_i)) \quad (1)$$

$$(x,y) = \mathrm{Inf}\,[T_P(x,y), \mathrm{Inf}_{u,v\in\gamma}(T_P(x,u)+1/2w_0+T_P(v,y)+ T_{NA}\text{-Length}(\gamma_{u,v}))]\} \quad (2)$$

According to this model, the *Shuttle-Time* function T_{NA} allows us to compute the expected running times of a shuttle which travels along a new route γ. The *No-New-Time* function T_P provides us with the expected transportation times of a user which moves through the network G without using this new route. So, the quantity $t^\gamma(x,y)$ provides us with an approximation of the expected transportation time of a user which moves from vertex x to vertex y and which uses in an optimal way a shuttle which travels along a new route γ according to a frequency w_0, under the hypothesis that this user expects his waiting time to be equal to $1/2w_0$. Consequently, the quantity $D_i^*.K(T_{NA}^*(x_i,y_i),t^\gamma(x_i,y_i))$ provides us with an evaluation of the demand for an access to the new shuttle route which is likely to emanate from users willing to go from origin x_i to destination y_i.

As was told during the introduction, few network transportation design problems involve non fixed demands (see for reference on vehicle routing problems: [2] [9] [23-24]. We can mention contributions of Florian [12] and Gartner [25] to *optimal traffic assignment problems*, and also from Leblanc [11] to *mode split assignment problems*. This is partially due to the difficulty related to the estimation and to the prediction of time dependent origin-destination flows (see for instance [10]). The **DMC** problem itself is somewhat new, and its formulation gives rise to several remarks:

- The **DMC** model clearly contains a lot of approximations: thus we should focus on avoiding "poor" solutions;
- A first analysis of the problem shows us that whatever be the kind of neighbourhood structure which will be used, a local search heuristic will have to cope with the existence of a large number of poor local optima;
- Dealing with time dependent demands means here dealing with the complex quantity *Global-Demand*(γ). Since the **DMC** problem is one of the basic components of a more general problem, we must control the computational costs related to the management of this quantity.

If we were focusing on mathematical precision, we would try do deal with the above **DMC** problem while using decomposition scheme for non linear optimization (Lagrangean, Benders or Proximal decomposition: see for instance [26-28]), or multicommodity flow models: see for instance [26] [29-31]. But, because of the above remarks, the methods which we are going to propose here will essentially aim at producing acceptable solutions while avoiding excessively high computational costs, and they will be based upon the introduction of a rewriting mechanism involving auxiliary criterion families. This approach might be compared with the *large neighbourhood* techniques used by [22] [32-33], or with the *smoothing* techniques used by [19]. The following sections 3 and 4 will be devoted to a general description of what we shall call *Pursuit Control* Approach, while section 5 will be devoted to a description of comparative numerical experiments.

3. BASIC TOOLS FOR AN ALGORITHMIC TREATMENT OF THE DMC PROBLEM : DOMAINS, NEIGHBOURHOODS AND APPROXIMATION SCHEME.

3.1. A Generic formulation of the DMC(k) Problem.

The **DMC**(k) problem admits a generic formulation **P** : {*Search for an object γ in a domain Ω which satisfies a constraint $C(\gamma)$ and which maximizes a quality criterion $\Phi(\gamma)$*}, where C is some constraint, i.e some expression whose evaluation yields a Boolean value. We introduce here, while keeping ourselves inside a generic optimization framework, the main algorithmic tools which we are going to use in order to deal with our problem.

We choose to define the domain Ω as the domain **Ω-GEOD**(k) of the circuits γ which may be written as records of:

- Some integer $N \geq 2$;
- A concatenation of N geodesic paths $\gamma_0 .. \gamma_{N-1}$ such that :
  - For any i = 0..N-1, Extremity(γ_i) = Origin(γ_{i+1}), i+1 being computed modulo N ;
  - $\sum_{i=0..N-1} T_{NA}\text{-Length}(\gamma_i) \leq k$.

The choice of this representation, which may look more complicated than the standard representation of γ as a circuit T_{NA}-Length no more than k, comes from the following intuition: the quality of a feasible circuit γ, will strongly depend here on its ability to connect in a fast way a few number of key vertices. This intuition, which is confirmed by both usual life and an analysis of simple case instances of the **DMC**(k) problem (one may, for instance, try to study what is happening when the network G is a 2-dimensional grid), suggests us that efficient **DMC** routes are likely to be circuits which may be decomposed according to the definition of the objects of **Ω-GEOD**(k), with a small related value N (N = 2..5).

3.2. Local Transformation Operators.

Designing an algorithm for the above generic problem **P** usually means defining *Local Search* operators : such an operator TL acts on an object γ of Ω, through some parameter value t which belongs to some domain $U_{TL}(\gamma)$, and it turns γ into an object $TL(\gamma)(t)$ of Ω. Two objects γ and γ' of Ω are *TL-neighbors* if there exists a value t in $U_{TL}(\gamma)$ such that $\gamma' = TL(\gamma)(t)$. We shall use here an operator **TL-HOM**, which acts on an object $\gamma = (N, \gamma_0..\gamma_{N-1})$ of **Ω-GEOD**(k) according to one of the following modes :

- Incrementation of N and insertion of a new geodesic single vertex path $\gamma_N = \{x_0\}$ into the sequence $\{\gamma_0..\gamma_{N-1}\}$;
- Replacement of the vertex $x_{i+1} = y_i$ by a vertex z adjacent or equal to $x = x_{i+1} = y_i$ and replacement of both geodesic paths γ_i and γ_{i+1} by two other geodesic paths which connect respectively x_i to z and z to y_{i+1} in such a way that the T_{NA}-Length of the resulting circuit does not exceed the value k.

The choice of this operator **TL-HOM** derives from the analysis of simple instances of the **DMC** problem, related to 2-dimensional grids or to triangulated planar graphs. In such cases, it will happen that any initial solution can be turned into an optimal solution through some well driven sequence of applications of the **TL-HOM** operator.

3.3. Pseudo-Random Approximating Functions.

Part of the difficulty of the **DMC** problem comes from the fact that the complexity of the *Global-Demand* quantity. Its management involves updating (after application to the current object γ of the operator **TL-HOM**) and *a priori* testing (testing the impact of an application of **TL-HOM** to the current object γ, before effectively applying this operator to γ). Besides, one can check, by studying some simple case examples (for instances examples related to grids) that the **TL-HOM** operator may induce the existence, for the *Global-Demand* function, of a large set of locally optimal solutions. In order to cope with those two difficulties, we are going to introduce auxiliary performance criteria, which will simultaneously work as **approximations** and as **pseudo-random perturbations** of the original criterion *Global-Demand*. While we shall wait until Section 4 in order to detail the way such a family of auxiliary functions will find its place inside a generic **PURSUIT CONTROL** algorithmic scheme, we already may guess that these functions will have to be definitely less time consuming than the original *Global-Demand* function, while statistically behaving in a similar way in relation to the **TL-HOM** operator. According to this purpose, we are going to define those auxiliary criteria while taking into account the following remarks:

- The quality of γ will strongly depend of its ability to induce fast connections between those of its vertices x and y which are close to high-level demand origin/destination vertices;
- The quality of γ will grow as γ is going to contain arcs located on a large number of short paths connecting high-level demand pairs (x_i, y_i), i in I, of the set OD.

In order to make those auxiliary criteria act like **pseudo-random** perturbations of the original criterion *Global-Demand*, we shall make them depend on a parameter vector λ which will be managed through random sorting.

So, we associate, with any circuit γ in **Ω-GEOD**(k), auxiliary quantities *Aux-Quality*$_\lambda(\gamma)$, which define what we call a **Pseudo-Random Approximating Function Family**, and which depend on a control parameter vector $\lambda \in \Lambda$, according to the following equations :

For any $\lambda \in \Lambda$, *Aux-Quality*$_\lambda(\gamma) = \Sigma_{x,y \text{ in } J\lambda(\gamma)} Q^{\lambda}_{x,y}. K_\lambda(T_{NA}{*}(x,y), T_{NA}\text{-Length}(\gamma_{x,y}))$

where:

- Every function K_λ is an *elasticity* function from R^{+2} to [0, 1], which linearly depends on the vector λ, and which is such that :
  - o $K_\lambda(T_{NA}{*}(x,y), T_{NA}\text{-Length}(\gamma_{x,y})) = 1$ iff $T_{NA}{*}(x,y) = T_{NA}\text{-Length}(\gamma_{x,y})$;
  - o $K_\lambda(T_{NA}{*}(x,y), T_{NA}\text{-Length}(\gamma_{x,y}))$ decreases and converges to 0 when $T_{NA}\text{-Length}(\gamma_{x,y})$ increases, $T_{NA}{*}(x,y)$ remaining fixed;
- The coefficients $Q^{\lambda}_{x,y}$ only depend on x and y and on the control vector λ: they provide an evaluation of the closeness of x and y in relation to the highest level demand pairs (x_i, y_i), i in I;
- $J_\lambda(\gamma)$ is an *ad hoc* set of vertex pairs of γ ;

Testing the behaviour of the pseudo-random approximating functions *Aux-Quality*$_\lambda$.

It is not easy to formalize this notion of pseudo-random approximation, which means here the ability of the low computing cost quantities *Aux-Quality*$_\lambda(\gamma)$ to simultaneously behave as approximations and random perturbations of the original criterion *Global-Demand*(γ). But we may test this concept in an experimental way. We did it on quasi-planar networks G = (X, E) with 100 vertices, 300 arcs, 200 origin/destination pairs, which are such that, for any value of the parameter k, the domain **Ω-GEOD**(k) is connected for the **TL-HOM** neighbourhood relationship. Then, while using:

- a value of the parameter k which is located between 1and 3 times the value of the T_{NA}–diameter of the network G;
- a piecewise linear elasticity function K ;

we have been performing the following experiment :

Randomly generate a set Σ of circuit pairs (γ, γ') of **Ω-GEOD**(k), which are **TL-HOM**-neighbors, and some set Λ_o of positive values of the control vector λ > 0 ;

For any pair γ, γ' in Σ and any λ in Λ_o, compare the signs of (*Global-Demand*(γ) – *Global-Demand*(γ')) and (*Aux-Quality*$_\lambda$(γ) - *Aux-Quality*-$_\lambda$(γ')).

Then we got the following results:

- **Test 1**: in 82% of the cases, λ being fixed, (*Global-Demand*(γ) – *Global-Demand*(γ')) and (*Aux-Quality*$_\lambda$(γ) - *Aux-Quality*-$_\lambda$(γ')) have identical signs, with a small dependency on the choice of λ. That means that *Global-Demand* and *Aux-Quality*$_\lambda$ behave in a similar way when submitted to an optimization process involving the local operator **TL-HOM**: thus the functions *Qualité-Aux*$_\lambda$, λ in Λ, behave like an **approximating family** for the *Global-Demand* criterion.
- **Test 2** : we get similar results when we fix the pair γ, γ', with a small dependency on the choice of the pair γ, γ': that means the functions *Qualité-Aux*$_\lambda$, λ in Λ, behave like **pseudo-random perturbations** of the *Global-Demand* criterion, when submitted to an optimization process involving the local operator **TL-HOM**.

3.4. A small scale instance exact method BENCH.

In order to provide a benchmark for a heuristic testing process, we implemented a branch and bound algorithm **BENCH**, which aims at solving **DMC** for small scale instances and whose design comes as follows :

- a **recursive representation** of the **DMC**(k) problem is obtained by considering, for any pair of disjoint arc subsets E_I and $E_F \subset E$, the sub-problem **DMC**$(k)_{EI, EF}$ which is induced by requiring γ to contain every arc of E_I and to contain no arc of E_F. Then *branching the* **DMC**$(k)_{EI, EF}$ sub-problem means picking up some arc e in $E – (E_I \cup E_F)$ and trying both cases : $e \in E_I$ and $e \notin E_F$.
- Bounding the **DMC**$(k)_{EI, EF}$ problem can then be done by computing, for any pair (x_i, y_i), I in I, some shortest path (in a sense which mixes T_{NA} et T_P), which uses no *shuttle* arc of E_F. While denoting $T_i(E_I, E_F)$, the length of this shortest path, we obtain an upper bound $V_{EI, EF}$ of the optimal value of the **DMC**$(k)_{EI, EF}$ problem by setting : $V_{EI, EF} = \Sigma_{i \text{ in } I} D_i^*. K(T_{NA}^*(x_i, y_i), T_i(E_I, E_F))$.

This algorithm **BENCH** does not enable us to deal with large or even medium scale instances, since it is based on a bounding process which is not powerful enough. Still, **BENCH** may be applied to small instances in order to provide us (Section 5) information about the behaviour of the heuristic scheme which will be described all throughout the section 4.

3.5. Two simple local search algorithms SIMPLE and SIM-SIMPLE.

The **SIMPLE** heuristic performs a simple **TL-HOM** based descent process. The **SIM-SIMPLE** performs a **TL-HOM** based *Simulated Annealing* process: the related *Temperature* parameter τ decreases from 1 to 0 in an arithmetic way (in 20 steps if we refer to the tests of the experimental section 5); for every value of the parameter τ, **TL-HOM** is tried M times, where M is a parameter of the process (M = 50 for most tests of Section 5), while the probability for the process to accept a non improving application of **TL-HOM** is equal to τ.

4. REWRITING APPROXIMATION SCHEME AND PURSUIT METAHEURISTICS

4.1 A PURSUIT Process

Let us now go back to our generic optimization problem **P**. We suppose that we are provided, in order to deal with **P**, with a local operator TL. We shall talk about a ***Rewriting Scheme*** if **P** may be reformulated through a family of problems $\{\mathbf{P}_\lambda, \lambda \text{ in } \Lambda\}$, globally equivalent to **P** in some sense. For instance:

- We shall talk about a *Strong Rewriting Scheme* if for any λ in Λ, the local (for the TL operator) and global optimal solutions of **P** and $\mathbf{P}_\lambda$ are the same.
- We shall talk about a **Probabilistic rewriting Scheme** if Λ is an event domain endowed with a probability measure such that maximizing $\Pi(\gamma)$ = (Probability for γ in Ω to be an optimal solution of $\mathbf{P}_\lambda$) means solving a problem which admits the same TL-local optima and global optima as **P.**

Provided with such a *Rewriting Scheme* **P** -> $\{\mathbf{P}_\lambda, \lambda \text{ in } \Lambda\}$, and with a TL based algorithmic process σ which takes in input a vector value λ in the control domain Λ, an initial solution γ in Ω, and which compute, in the search domain Ω, a TL-local optimum γ' of $\mathbf{P}_\lambda$, we may use the parameter λ in Λ as a **Control Parameter** and deal with the problem **P** according to the following algorithmic scheme, which will be called **PURSUIT** scheme (the *pursuer* parameter λ tries to catch the *evader* γ) :

Scheme 1: PURSUIT scheme

Initialize λ in Λ; **Not** Stop;
Initialize γ in Ω;
While Not Stop **do**
  Locally Solve the problem $\mathbf{P}_\lambda$, through application of the σ local search process to γ; (**I0**)
  Let $\gamma' = \sigma(\gamma,\lambda)$ in Ω be the resulting object;
  Depending on the characteristics of γ', **Replace or not** γ by γ'; (**I1**)
  Update Stop; **If** Not Stop **Then Generate** another value λ;

The above **PURSUIT** process takes in a straightforward way advantage of the rewriting scheme **P** -> $\{\mathbf{P}_\lambda, \lambda \text{ in } \Lambda\}$. It works like a local search process, while involving a topology (neighbourhood structure) REG(σ) which is defined in an implicit way on the domain Ω by : two objects γ and γ' of Ω are REG(σ)-neighbours iff there exists λ in Λ such that $\gamma' = \sigma(\gamma, \lambda)$. In the case when σ is itself some local search process, related to some topology r, the REG(σ) topology appears at the same time as a **regularization** and as an **amplification** of r. One may compare this approach to the smoothing approach which was proposed by [19], or to the large neighbourhood approach developed by [4] or [32].

4.2 PURSUIT Scheme, Pseudo-Random Approximating Functions and the DMC Problem.

Let us keep on with our generic problem **P**, which we supposed to be given together with some local transformation operator TL. Coming back to the pseudo-random approximating function families of Section 3, we are going to study the way such a family can be used in order to provide us with a kind of convenient rewriting scheme. According to this prospect, we define:

Pseudo-random approximating function family for the problem P.

It will be any family of real valued functions $\{\Phi_\lambda, \lambda \text{ in } \Lambda\}$, defined on the domain Ω, and such that:

- the following problems $\mathbf{P}_\lambda$, λ in Λ: {*Search for γ in Ω such that $C(\gamma)$ and that $\Phi_\lambda(\gamma)$ is the largest possible*}, may be viewed as defining a probabilistic rewriting scheme of **P**;
- computing a local solution of $\mathbf{P}_\lambda$ through the application of a TL-local search process σ induces the same kind of computational costs as testing the effect of an application of the TL operator on the criterion value $\Phi(\gamma)$. In such a case, we say that the resulting topology REG(TL) which is defined by : γ

REG(TL) γ' iff there exists λ such that it is possible to get the local solution γ' of $\mathbf{P}_\lambda$ by applying of σ from γ, is the *regularization of the TL topology through the family {Φ_λ, λ in Λ}.*

The above formulation « *may be viewed* » is voluntarily ambiguous. Formalizing it would be possible. But it will happen that, in any practical application, the $\mathbf{P}_\lambda$ problem family is going to be defined in an empirical way. Intuitively, what we are expecting from the function family {Φ_λ, λ in Λ} is that, for any λ in Λ:

- the computational costs related to the management of Φ_λ are smaller (in a significant way) than similar computing costs induced by Φ.
- Φ_λ simultaneously behave as an approximation and as a perturbation of Φ.

Then it will come that using the related regularization REG(TL) of the TL local operator will help us in driving the object γ until promising areas, without requiring an increase of the computing costs.

Casting the DMC Problem into the PURSUIT framework.

Clearly, in the case of the **DMC** problem, the functions *Aux-Quality*$_\lambda$ which we introduced in Section 3 are going to play the role of a Pseudo-random approximating function family: so we consider the quantities *Aux-Quality*$_\lambda$(γ) as pseudo-random approximations of the quantities *Global-Demand*(γ), and we do as if the problems **CDM**(k)$_\lambda$:

{Search for a circuit γ in **Ω-GEOD**(k) which maximizes the auxiliary criterion *Aux-Quality*$_\lambda$(γ)},

were defining a probabilistic rewriting scheme for the **DMC**(k) problem. Depending on the strategy we may want to apply when performing a local search process according to the regularized topology REG(**TL-HOM**), this yields several algorithms. Let us propose three of them, which will be tested in the following section 5.

Scheme 2: RANDOM-PURSUIT scheme, a random walk version of the generic PURSUIT scheme.

Parameter: a control parameter K1 ;
Process:
Initialize γ; I <- 1 ;
Solcour <- γ ; Valcour <- *Global-Demand*(γ) ;
While I $\le$ K1 **do**
Randomly Generate λ ; I <- I+ 1 ; γ' <- γ ; (**3**)
While t exists in $U_{TL\text{-}HOM}$(γ') such that
Aux-Quality-$_\lambda$(T(γ')(t) > *Aux-Quality*$_\lambda$(γ') **do** γ' <- TL-HOM(γ')(t);
γ <- γ' ;
If *Global-Demand*(γ) > Valcour
Then {Solcour <- γ;
Valcour <- *Demande-Globale*(γ)}.

Running **RANDOM-PURSUIT** means driving the object γ along some kind of random walk on the domain **Ω-GEOD**(k): the involved transitions derive from the regularization of the **TH-HOM** topology through the pseudo-random approximating function family {*Aux-Quality*$_\lambda$, λ in Λ}. The (**3**) instruction is performed through a random sorting process which enables us to enlarge in a powerful way the moves of γ.

Scheme 3: DESCENT-PURSUIT scheme, a descent version of the generic PURSUIT scheme.

Parameters: a control parameter K2.
Process :
Initialize γ; **Not** Stop ;
While Not Stop **do**
Stop; I <- 1 ; **Not** Stop1;
While Not Stop1 **and** I $\le$ K2 **do**

Randomly Generate λ ; I <- I+ 1 ; γ' <- γ ; (**3**)
While t exists in $U_{TL\text{-}HOM}(\gamma')$ such that
Qualité-Aux$_\lambda$(T(γ')(t) > *Qualité-Aux*$_\lambda$(γ') **do**
γ' <- TL-HOM(γ')(t);
γ <- γ' ;
If *Demande-Globale*(γ') > *Demande-Globale*(γ)
Then {γ <- γ'; Stop1; **Not** Stop}.

Running the **DESCENT-PURSUIT** process means driving the object γ throughout the search domain **Ω-GEOD**(k) according to a simple descent control, while using transitions which derive from the regularization of the **TH-HOM** topology through the pseudo-random approximating function family {*Aux-Quality*$_\lambda$, λ in Λ}.

Scheme 4: SIM-PURSUIT scheme, a Simulated Annealing version of the generic PURSUIT Scheme.

Parameters :
- A function K12 which associates, with any **temperature value** τ in [0,1], some iteration number K12(τ) ;
- An increasing function V12 which, with any **temperature value** τ in [0,1], associates an **acceptation threshold** value V12(τ) in [0,1] in such a way that V12(0) = 0 et V12(1) = 1;
- A decrementation step δ.

Process :
Initialize γ; τ = ***Initial Temperature*** <- 1 ;
Solcour <- γ ; Valcour <- *Global-Demand* (γ);
While $\tau \neq 0$ **do**
I <- 0 ;
While I $\leq$ K12 **do**
Randomly Generate λ; I <- I+ 1; γ' <- γ ; (**3**)
While t exists in $U_{TL\text{-}HOM}(\gamma')$ such that
Aux-Quality$_\lambda$(T(γ')(t) > *Aux-Quality*$_\lambda$(γ')
do γ' <- **TL-HOM**(γ')(t);
Randomly Generate V in [0,1] ;
If *Global-Demand*(γ') > *Global-Demand*(γ') **or** (V $\leq$ V12(τ)) **Then** γ <- γ' ;
If *Global-Demand*(γ) > Valcour **Then**
{Solcour <- γ; Valcour <- *Global-Demand*(γ)} ;
Decrementation of τ: τ <- $\tau - \delta$;

Running **SIM-PURSUIT** means driving the object γ throughout the search domain **Ω-GEOD**(k) according to a simulated annealing control, while using transitions which derive from the regularization of the **TH-HOM** topology through the pseudo-random approximating function family {*Aux-Quality*$_\lambda$, λ in Λ}.

Initializing γ when launching the processes **SIMPLE**, **SIM-SIMPLE**, **RANDOM-PURSUIT**, **DESCENT-PURSUIT** and **SIM-PURSUIT** is done through successive applications of a construction operator **S** in such a way that at any time during this construction process, we choose the parameter value for this operator **S** as if we were trying to maximize a quantity *Aux-Quality*$_\lambda$ *(γ*)*}. That means that we follow the guidelines of the general **PURSUIT** framework in order to provide a **non deterministic** greedy version **GREEDY-PURSUIT** of the generic **PURSUIT** scheme. This non deterministic algorithm is applied a number K3 of times, and this number K3 becomes a control parameter of the initialization procedure **GREEDY-PURSUIT**.

4. NUMERICAL EXPERIMENTS

We have been performing **two classes of experiments**:

- The first one (Section 5.1) is devoted to medium scale networks and it aims at providing us with a comparative analysis of the behaviour of the algorithms **SIMPLE**, **SIM-SIMPLE**, **DESCENT-PURSUIT**, **RANDOM-PURSUIT**, **SIM-PURSUIT**, **GREEDY-PURSUIT** of sections 3 and 4.
- The second one (Section 5.2) is devoted to small scale networks and it aims at providing us with an evaluation, for such networks, of the quality of the results which are computed by the above-mentioned heuristics. It involves comparisons with the results produced by the exact method **BENCH** of Section 3.

5.1 A comparative analysis related to medium scale networks.

We have been working here on networks G = (X, E) with n = 100 vertices, m = 300 arcs, I = 200 origin-destination pairs. We tested, while using such randomly generated networks, the behaviour of the descent procedure **SIMPLE** and of the simulated annealing procedure **SIM-SIMPLE-REC**, (see Section 3), both based upon the use of the **TL-HOM** local search operator on the original criterion *Global-Demand*, and we compared it to the behaviour of the **PURSUIT** procedures **RANDOM-PURSUIT**, **DESCENT-PURSUIT** and **SIM-PURSUIT** of Section 4. While doing it, we focused on both the computational times and on the quality level of the solutions which were obtained: that means that we kept the memory, every time we tested a procedure on a given instance, of the value **VAL** which provides us with the ratio between the value *Global-Demand*(γ) of the solution γ which was obtained through this procedure and the best value we could obtain on the same instance, as well as of the following quantities related to the computational costs:

- The number **NΦ** of times a quantity *Global-Demand*(γ) is computed during the execution of the procedure;
- The numbers **NTLΦ** and **NTLΦ**$_\lambda$ of times an evaluation of the impact on the quantities *Global-Demand*(γ) and *Aux-Quality*$_\lambda$(γ) of an application of the **TL-HOM** local operator is performed;
- The CPU time **CPU** which was required by the procedure in order to run the given instance;

We used the following Hardware and Software supports: C++ programming language C++ ; Unix SUN Sparc 10 working stations.

Then we got results which could be summarized inside the table 1 below, which correspond to the value K3 = 1 of the diversification parameter: every procedure **SIMPLE**, **SIM-SIMPLE**, **DESCENT-PURSUIT**, **RANDOM-PURSUIT** and **SIM-PURSUIT** is launched exactly once for every instance, while starting from the initial solution computed by **GREEDY-PURSUIT**.

Table 1: A comparative study.

	VAL	NΦ	NTLΦ	NTLΦ$_\lambda$	CPU (s)
GREEDY-PURSUIT	0.64	1	0	0	3
SIMPLE	0.72	2	205	0	94
SIM-SIMPLE	0.85	2	1000	0	497
RANDOM-PURSUIT	0.86	41	0	3480	128
DESCENT-PURSUIT	0.88	63	0	3860	160
SIM-PURSUIT	0.92	200	0	13850	758

Not surprisingly, **SIMPLE** and **SIM-SIMPLE** do not yield good results, due to the lack of power of the neighbourhood topology defined by **TL-HOM**. For such a local optimum γ, there exist many values λ such that γ is not a local optimum for *Aux-Quality*$_\lambda$. Thus, performing a descent process from γ while using the auxiliary criterion *Aux-Quality*$_\lambda$ get us out deadlock situations. It comes that both processes **RANDOM-PURSUIT** and **DESCENT-PURSUIT** yield better results than **SIMPLE**, and induce computational costs which are far less than those induced by **SIM-SIMPLE**. As a matter of fact, one can check that the computing costs related to a trial of the **TL-HOM** operator to *Global-Demand* are twice hundred times more than those related to a trial of the same operator to *Aux-Quality*$_\lambda$. It means that using the **PURSUIT** rewriting scheme together with the pseudo-random approximating functions *Aux-Quality*$_\lambda$, provides us with an « intelligent » strategy for the management of enlarged neighbourhoods, and this without inducing any significant growth of the computational costs. In a not surprising way, it is the mixed procedure **SIM-PURSUIT** which yields the best results while the greedy non deterministic procedure **GREEDY-PURSUIT**, which is efficient as an initialization/diversification procedure, scarcely succeeds in yielding efficient solutions by itself.

The fact that the computational times are high enough comes from the fact that our codes have not been optimized.

5.2 Impact of the Diversification parameter K3.

The following table 2 provides us with an evaluation of the influence of the initialization process **GREEDY-PURSUIT** on the quality of the whole local search processes. Every procedure **SIMPLE**, **SIM-SIMPLE**, **DESCENT-PURSUIT**, **RANDOM-PURSUIT** and **SIM-PURSUIT** is launched exactly K3 = 5 times for every instance, while starting every time from an initial solution computed by **GREEDY-PURSUIT**.

Table 2: impact of the diversification parameter K3

	VAL
GREEDY-PURSUIT	0.73
SIMPLE	0.79
SIM-SIMPLE	0.94
RANDOM-PURSUIT	0.96
DESCENT-PURSUIT	0.97
SIM-PURSUIT	1.00

Diversification clearly improves our results. Since the above local search procedures are sensitive to the way initialization is performed, it is likely that, in many cases, we did not obtain the best **DMC** solution.

5.3 Impact of the control parameters K1 and K2

The table 3 below provides us with an evaluation of the influence of the behaviour of RANDOM-PURSUIT an DESCENT-PURSUIT of both control parameters K1 and K2. We try here the values K1 = 40, 100, 200 and K2 = 20, 40, 80, while the diversification parameter K3 remains equal to 1.

Table 3: impact of the control parameters K1 and K2

	VAL	**NΦ**	**NTLΦ$_\lambda$**	**CPU**
RANDOM-PURSUIT(40)	0.86	41	3480	128
RANDOM-PURSUIT(100)	0.93	101	7500	302
RANDOM-PURSUIT(200)	0.97	201	14200	588
DESCENT- PURSUIT (20)	0.88	63	4560	160
DESCENT- PURSUIT (40)	0.93	112	7800	308
DESCENT- PURSUIT (80)	0.95	196	13400	597

As expected, giving more time to the search process improves the results, without ensuring us to get the best solution.

5.4 Comparison with the exact method BENCH in the case of small networks.

We have been working here on small scale networks with n = 30 vertices, m = 90 arcs and I = 50 origin-destination pairs, which, in most cases, allowed us to compute an optimal solution through the branch and bound procedure **BENCH**. Then we focused on precision, and computed, for a 10 instance package and for both procedure **BENCH** and **SIM-PURSUIT**, the following quantities:

- **VALUE** = ratio between the best value which could be computed by the tested procedure and the theoretical optimal value (obtained through the **BENCH** procedure);
- **EMAX** = the worst ratio which was obtained by the tested procedure;
- **RATE** = the number of times the tested procedure could compute an optimal solution.

Then we get results which can be summarized as follows.

Table 4: A small scale instance analysis

	VAL	**RATE**	**EMAX**	**CPU(s)**
SIM-PURSUIT	0.98	8	0.87	125
BENCH	1.00	10	1.00	3250

In most cases, **SIM-PURSUIT** was able to get an optimal solution.

6. CONCLUSION

The main purpose of this paper was to study an original network design transportation problem which involved elastic demands. This very specific feature made arise difficulties related to the management of complex quantities, whose computation tended to be excessively time consuming. Trying to handle those difficulties led us to introduce new metaheuristic scheme based upon the used of what we called *pseudo-random approximating function families*. This scheme, which involves a kind of rewriting technique, aims at simultaneously making the computational costs decrease and regularizing the topology of the search domain. We tested it on academic instances, and we got satisfactory results.

We should now try to go ahead with this study while working at two levels:

- The application level: we are attending the emergence of a class of innovative transportation services, which involve shared vehicles, autonomous vehicles… Relevant demand models and efficient related network design software should help in ensuring the economic viability of those services.
- The level of the algorithm design methods: it should be interesting to go further with the scheme of Section 4, and to study the way pseudo-random approximating function families can be defined.

ACKNOWLEGMENTS

Authors thank a lot the organizers of the IEEE International Conference LT'2007, which was held in 2007 in SOUSSE, TUNISIA.

REFERENCES

1. CONSTANTIN I., *L'optimisation des fréquences d'un transport en commun*, Rapport CRT 881 (PHD Thesis, Dir. M. FLORIAN), Université de MONTREAL, 1993.
2. CORDEAU J.F., LAPORTE G., *The Dial a Ride problem: variants, models and algorithms*, *40R* 1-2, p 89-101, 2003.
3. DEITCH R., LADANY P., *The one period bus touring problem solved by an efficient heuristic*, *EJOR* 127, p 69-77, 2000.
4. FEILLET D., *Problèmes de tournées avec gains: étude et applications au transport inter usines*, Thèse de Doctorat, Ecole Centrale PARIS, 2001.

5. GENDRON B., *Modèles et algorithmes pour problèmes d'optimisation de réseaux avec coûts fixes*, Rapport CRT 95-21, Université de Montréal, 1995.
6. MAGNANTI T., *Combinatorial optimization and vehicle fleet planning perpectives and prospects*, Networks 11, p 179-214, 1981.
7.MITROVIC S., KRISHNAMURTI R., LAPORTE G., *Double horizon based heuristics for the dynamic pickup and delivery problem with time windows*, Transportation Research B, 35, 1, 2003.
8. PUCHTA M., GOTLLIEB J., *Solving car sequencing problems by local optimization*, Appl. Evolutionary Comp., LNCS 2279, p 132-142, 2002.
9. TOTH P., VIGO V., *The vehicle routing problem*, SIAM Monographs on Disc. Maths and Appli. , 2002.
10. ASHOK K., *Estimation and prediction of time dependent origin-destination flows*, PHD Thesis (Dir. M. BEN-AKIVA), MIT, 1998.
11. LEBLANC L.J., FARHANGIAN K., *Efficient algorithms for solving elastic demand traffic assignment problems and mode split assignment problems*, Transp. Sci. 15, p 307-317, 1981.
12. FLORIAN M., *An introduction to network models used in transportation planning*, in M. FLORIAN Ed, Transp. Plann. Models, North Holland, Amsterdam, p 137-152, 1984.
13. WARDROP J., *Some theoretical aspects of road traffic research*, Proc. of the Institute of Civil Engineering, II, 1, p 325-378, 1952.
14. QUILLIOT A., YON L.,*Tournées à demandes élastiques: une métaheuristique basée sur une expression polymorhique du problème*; Proceeding Conf *LT07*, SOUSSE, TUNISIE, 6 pages, 2007.
15. DREYFUS C, SIARRY P., *La Méthode du Recuit Simulé*, IDEST PARIS, 1988.
16. KIRKPATRICK S., GELATT C., VECCHIT M., *Optimization by simulated Annealing*, RC 9355, 4/2/82, Computer Science and Engineering Techniques, IBM T.J. Watson Center, N.Y., 1982.
17. GLOVER F., *Tabu Search*, ORSA Jour. of Comp. 1-3, p 190-206, 1989.
18. DE WERRA D., HERTZ A., *Tabu Search Techniques: A tutorial and an application to neural networks*, OR Spectrum 11, p 131-141, 1989.
19. J. GU, X. HUANG: "Efficient local search with search space smoothing : a case study of the travelling salesman problem"*; IEEE Trans. On Systems, Man, Cybernetics* 24, 5, p 728-736 (1994).
20. M.MARCHIONI: "On the expressive power of rewriting"; Proc FST/TCS 97, *LNCS* 1346, p 88-102, 1997.
21. A.NIEUWENHUIS: "Rewriting techniques and applications"; Proc 14 th RTA Conf. 2003, *LNCS* 2706, p 67-80, 2003.
22. M.PALPAN, C.ARTIGUE, P.MICHELON: "LSSPER : Solving the resource constrained project scheduling problem with large neighbourhood search » ; *Annals of Operat. Research* 31-1, p 237-257, (2004).
23. CORDEAU J.F., GENDREAU M., LAPORTE G., POTVIN J.Y., SEMET F., *A guide to vehicle routing heuristics*; Jour. Op. Res. Soc., 53-5, p 512-522, 2002.
24. GENDREAU M., LAPORTE G., MESA J.A., *Locating rapid transit lines; decision criteria and methodology*, Rapport CRT 907, Université de Montréal, 1993.
25. GARTNER N. H., *Optimal traffic assignment with elastic demands: a review*, Transp. Sciences 14, p 192-208 and 174-191, 1980.
26. AHUJA RV, MAGNANTI T.L., ORLIN J. B.: *Network Flows: Theory, Algorithms and Applications*, Prentice hall, Englewood Cliffs, N.J., 1993.
27. HOSSEIN P.A., BERTSEKAS D. P., TSENG P., *Relaxation methods for networks problems with convex arc costs*, S.I.A.M. Journ. On Control and Optimization 5, 25, p 1219-1273, 1987.
28. GAVISH B., *Topological design of telecommunication networks: local access design methods*, Ann. of Operat. Research 33, p 17-71, 1991.
29. GOFFIN J.L., GONDZIO J., SARKISSIAN R., VIAL J.P, *Solving non linear multicommodity flow problems by the analytic center cutting plane method*, Math. Prog. 76, p 131-154, 1997.
30. PASCHOS V. Ed: *Optimisation Combinatoire : Fondements et Méthodes*, Hermes, PARIS, 2005.
31. MINOUX M. *Network synthesis and optimum design problems :models, solution methods and applications*; Networks 19, p 313-360, 1989.
32. AHUJA R.K., ERGUN O., ORLIN J.B., UNNEN A.P., *A survey of very large scale neighbourhood search techniques*; Dis. Appli. Maths 123, p 75-102, 2002.
33. ESTELLON B., GARDI F., NOUIOUA K., *Ordonnancement de véhicules : une approche par recherche locale à grand voisinage*, Actes JFPC, p 205-220, 2005.

Received July 14, 2008