A GENERALIZED K-OPT EXCHANGE PROCEDURE FOR THE MTSP*

JEAN-YVES POTVIN, GUY LAPALME, JEAN-MARC ROUSSEAU

Centre de recherche sur les transports, Université de Montréal,
C.P. 6128; Succ. "A", Montréal, Quebec, Canada H3C 3J7.

ABSTRACT

This paper presents a technique that generalizes the classical k-opt exchange procedure for the symmetric MTSP (Multiple Traveling Salesmen Problem), by considering exchanges leading to the partition of a single tour into a number of smaller subtours. These subtours are then merged back together into an equivalent single tour. In this way, more exchange opportunities are explicitly considered during the k-opt procedure and greater improvements can be achieved. As we will show, the technique is particularly powerful for MTSP problems with time windows. Numerical results are presented at the end of the paper for the classical non-constrained MTSP and for the MTSP with time windows.

Key Words : Multiple Traveling Salesmen Problem, generalized k-opt exchange procedure, time windows.

RÉSUMÉ

Le présent article introduit une technique qui généralise la procédure classique d'échange k-opt pour le problème symétrique des M-voyageurs de commerce. Cette technique considère les échanges menant à la scission d'un tour unique en sous-tours multiples, sous-tours qui sont par la suite réunis en un tour unique équivalent. De cette façon, de nouvelles opportunités apparaissent au cours de la procédure d'échange permettant ainsi d'améliorer encore davantage la solution initiale. Tel que démontré dans l'article, la technique que nous avons mise au point est particulièrement efficace pour les problèmes avec fenêtres de temps. Des résultats numériques sont présentés à la fin de l'article pour le problème classique non-contraint des M-voyageurs de commerce et pour le problème avec fenêtres de temps.

Mots-Clés : Problème des M-voyageurs de commerce, procédure d'échange k-opt généralisée, fenêtres de temps.

1. INTRODUCTION

It is well known that the apparent generalization that the MTSP provides is somewhat illusory, since each problem of this type can be converted into an equivalent TSP, as described by Svestka & Huckfeldt (1973), Bellmore & Hong (1974). We get that equivalence by creating M copies of the depot $D1, \ldots, DM$, each connected to the other nodes exactly as was the original depot. The M copies are not connected or are connected by arcs with a length sufficiently large to prevent their use. Such a modification defines a new "augmented" network in which any MTSP solution can be represented as a TSP solution (see Figure 1).

Assuming a symmetric network, it is then possible to apply the classical k-opt exchange procedure on this unique tour, see Lin (1965), Lin & Kernighan (1973). This method defines a k-change of a tour as consisting of the deletion of k edges in the tour and their replacement by k other edges to form a new tour. A tour is then said to be k-opt if it is not possible to improve it via a k-change. Our method generalizes this well-known approach by explicitly considering "generalized" k-changes, that is, k-changes leading to the partition of the tour into a number of smaller subtours. For a

* Accepted. April 1989.

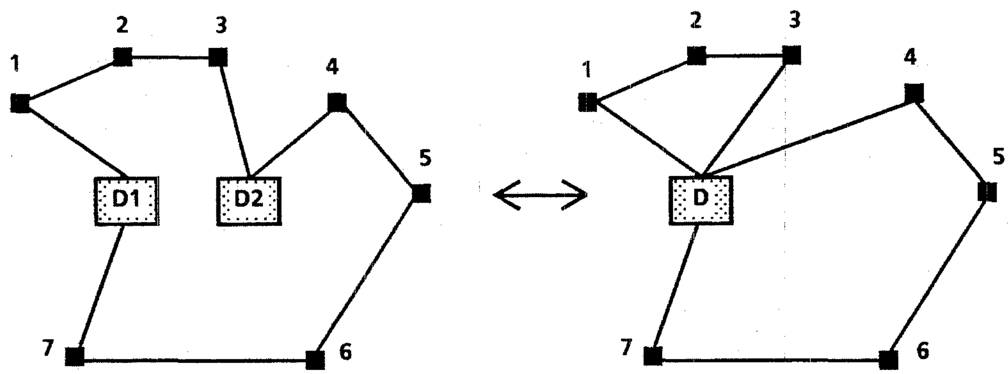


Figure 1. MTSP-TSP Equivalence

TSP, any partition of the tour leads to non feasible solutions, but for the MTSP, solutions with multiple subtours can be feasible. Indeed, a solution with multiple subtours is feasible if and only if at least one copy of the depot is included in each subtour. This observation has not been reported yet in the literature and this paper shows its potential benefits for symmetric MTSP.

2. THE GENERALIZED k-CHANGE

Figure 2 shows that there are three different ways to link two subchains together. Among these cases, case (b) is the only one explicitly considered by the classical 2-opt exchange procedure (leaving out case (a) which, in the context of an exchange procedure, restores the initial tour). Likewise, there are fifteen ways to link three subchains together and the classical 3-opt exchange procedure considers seven cases out of fifteen.

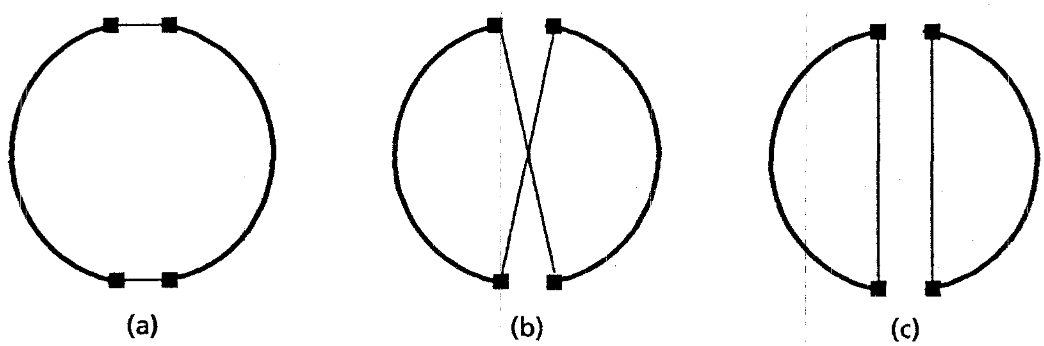


Figure 2. Three ways to connect two subchains

In the context of MTSP problems, however, solutions involving multiple subtours, like case (c) in Figure 2, are potentially interesting. Such a solution is produced by applying a generalized 2-change to the initial tour (a). We can easily deduce the feasibility or non feasibility of this new solution, as follows:

(a) if at least one copy of the depot is included in each subchain, then case (c) represents a new feasible solution.
(b) if one subchain has no copy of the depot, then case (c) is a non feasible solution.

Our generalized exchange procedure will thus consider all possible k-changes leading to new feasible solutions (that is, feasible solutions involving one tour or multiple subtours). For $k = 2$ or $k = 3$, this will at most double the number of possible exchanges. When a new solution is composed of multiple subtours, an equivalent unique tour is then easily regenerated by a judicious exchange of "equivalent edges" (i.e. edges connecting a given node to each copy of the depot). Figure 3 shows how an equivalent single tour can be produced with this simple technique.

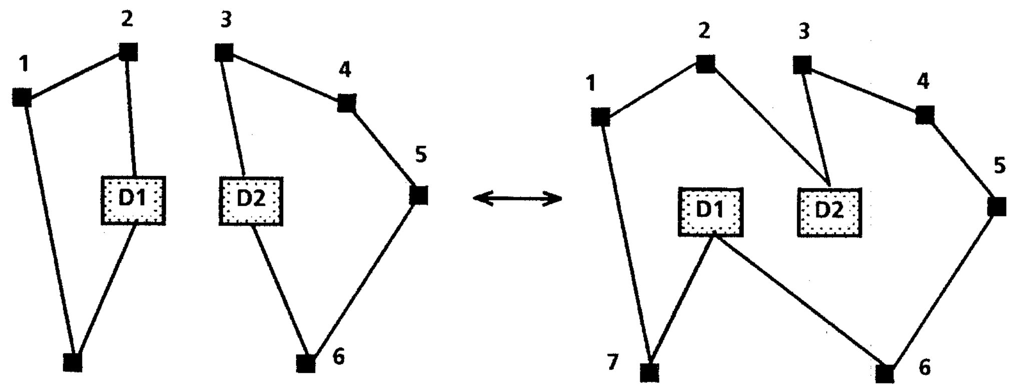


Figure 3. Regenerating a single tour

Following the terminology of Lin (1965), a tour is k-opt* if its length cannot be decreased via a classical or a generalized k-change. Statement (2.1) is then easy to prove, since the set of exchanges considered by the classical approach is included in the set of exchanges considered by our generalized approach:

Statement (2.1) *Any k-opt* tour is k-opt*

Figure 4 shows that the inverse of statement (2.1) is not true. In this example, the distance matrix has been defined in such a way that the replacement of edges $\{(2,3),(6,7)\}$ by edges $\{(2,7),(6,3)\}$ is the only way to reduce the length of the 2-opt tour T, so as to obtain a 2-opt* tour T*.

In Figure 4, we can see that four edges have been exchanged in order to get the 2-opt* tour (including the exchange of "equivalent" edges so as to reunify the two subtours into an equivalent unique tour). We can easily generalize this observation as follows:

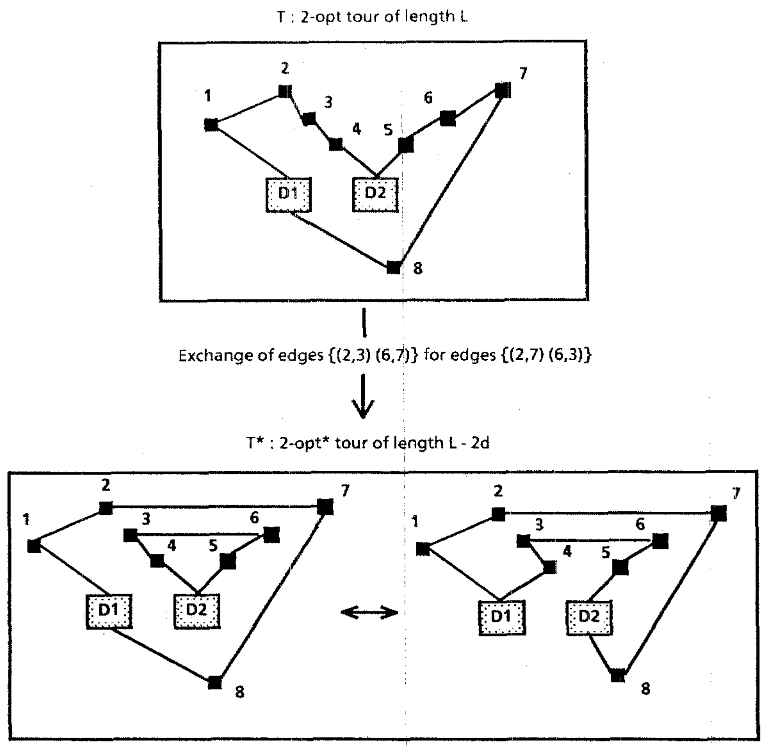
Statement (2.2) *A generalized k-change can lead to the substitution of at most* $2k$ *edges in a tour defined on the "augmented network". Moreover, if there are* $k + x$ $(x > 0)$ *substitutions then at least x of them involve equivalent edges.*

Obviously, a generalized k-change is more powerful than a classical k-change, since the latter cannot authorize more than k substitutions in any given tour. In fact, the generalized k-change neighborhood is a subset of the classical $2k$-change neighborhood.

3. THE GENERALIZED k-OPT EXCHANGE PROCEDURE

We can now define our generalized k-opt (k-opt*) exchange procedure as follows:

(1) Generate an initial solution
(2) Try to decrease the length of the tour using a classical or a generalized k-change
(3) If the exchange leads to multiple subtours, regenerate an equivalent unique tour
(4) Repeat steps 2 and 3 until no more improvement is possible.



d(i,j) = x $\quad\forall$(i,j) $\in$ E

d(i,j) = x + 2d $\quad\forall$(i,j) $\notin$ E and (i,j) $\notin$ {(2,7), (6,3)}

d(2,7) = d(6,3) = x - d

where E = {edges on T} $\cup$ {(4,D1) (5,D1) (1,D2) (8,D2)}

d > 0

Figure 4. 2-opt vs 2-opt* tour

Based on complexity results presented by Syslo, Deo & Kowalik (1983), the complexity of step 2 of this procedure is (n choose k) $\times$ $(((2k-1)\times(2k-3)\times\ldots\times 3\times 1)-1)$, which is equivalent to:

$$\binom{n}{k}\,(2^{k-1}(k-1/2)!-1)$$

where

$$n = \text{number of edges in the tour}$$
$$(k-1/2)! = (k-1/2)\times(k-3/2)\times\ldots\times 3/2$$

Step 2 of the equivalent k-opt exchange procedure (which only considers "classical" k-changes) gives:

$$\binom{n}{k}\,(2^{k-1}(k-1)!-1)$$

Hence, for any given k ($k > 1$), the complexity of step 2 for the k-opt* and k-opt exchange procedures are respectively of order

$$C'\binom{n}{k} \quad \text{and} \quad C\binom{n}{k} \qquad \text{where } C' > C.$$

Thereby, the complexity is in the order of n^k for both approaches.

For example, with $k = 2$ the complexity is in the order of n^2 with $C' = 2$ and $C = 1$. Hence, for the same number of iterations, the 2-opt* exchange procedure should take about twice as much time as the 2-opt procedure, independently of the number of edges in the tour (regenerating a single tour from multiple subtours can be neglected, since for a MTSP, the complexity is in the order of M, which is usually very small as compared with the number of edges in the tour). Notice, for comparison purposes, that a 3-opt exchange procedure requires on the order of n^3 computations.

4. THE MTSP WITH TIME WINDOWS

In the previous sections, we described our general approach for the "constraint-free" MTSP. However, the k-opt exchange procedure is also suitable for constrained problems. In such a case, after each exchange, an additional check has to be made in order to be sure that the new solution satisfies all the constraints.

Working with various constrained problems, we discovered that the general procedure is particularly powerful for solving MTSP problems with time windows. As illustrated in Figure 5, starting with a feasible solution, the classical 2-change is not very useful when nodes are shifted from one route to an other (a k-change can obviously shift nodes from one route to an other, when applied to an "augmented" network). As depicted in Figure 5, a 2-change links the first nodes of route 1 to the first nodes of route 2. This strategy is very bad because the first nodes in a feasible route usually have time windows with low upper bound (that's why they are serviced first!), while the last nodes have higher upper bound. Therefore, the 2-change is likely to produce a non feasible route because nodes 5 and 6, the first nodes of route 2, are shifted to route 1 and are now serviced after nodes 1 and 2. Since nodes 5 and 6 are likely to have time windows with low upper bound, the new route is probably non feasible.

On the other hand, the generalized 2-change is particularly well suited for this problem, since it links the first nodes of route 1 with the last nodes of route 2, and vice-versa (see Figure 5). This new solution is thus likely to be feasible (for similar reasons as those mentioned above). This is obviously a very interesting observation and the results presented in the next section will emphasize this point.

5. EXPERIMENTAL RESULTS

In order to evaluate the generalized k-opt (k-opt*) exchange procedure, we ran experiments with symmetric networks both in the context of "constraint-free" MTSP problems and in the context of MTSP problems with time windows. All experiments were run on the 1108 Lisp Xerox workstation. This machine was chosen because of the availability of a general routing software particularly well suited for our application, see Lapalme, Potvin & Rousseau (1988). However, because Lisp machines are symbolic processing machines (rather than "number crunching" machines), the computation times are slower than on standard hardware.

5.1 Classical "Constraint-free" MTSP

To illustrate the benefits to the k-opt* exchange procedure for the MTSP, we ran experiments using randomly generated networks with 20, 40, 60 and 80 nodes (ten networks for each size). We generated initial tours for two and three salesmen with a fairly good heuristic known as "farthest insertion", see Rosenkrantz (1977). We then applied in turn the 2-opt, 2-opt* and 3-opt exchange procedures to these initial tours (only one improvement procedure was applied to each initial tour). Table 1 shows for each network size and for each algorithm the total length of the final tours and the percentage of

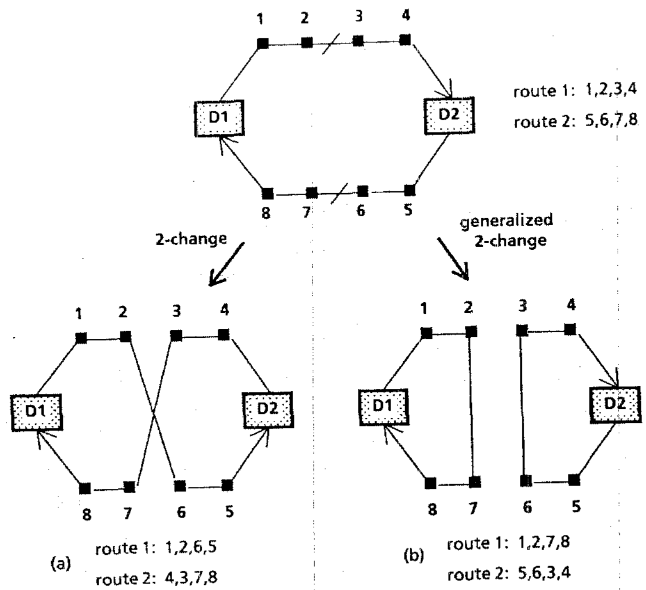


Figure 5. 2-change vs generalized 2-change

improvement over the initial solution (INIT). Computation times in seconds are shown within parentheses under each result and each one of them is the mean taken over ten problems.

Table 1a. Experimental Results for 2TSP

2TSP	20 nodes	% impr.	40 nodes	% impr.	60 nodes	% impr.	80 nodes	% impr.
INIT	2253.65		2895.81		3257.47		3874.01	
INIT +2OPT	2194.64 (5.3)	2.6	2828.13 (13.7)	2.3	3185.81 (42.8)	2.2	3777.15 (70.3)	2.5
INIT +2OPT*	2159.38 (13.3)	4.2	2803.14 (36.3)	3.2	3156.49 (106.1)	3.1	3746.66 (189.5)	3.3
INIT +3OPT	2088.85 (117.5)	7.3	2742.81 (595.6)	5.2	3071.79 (2634.6)	5.7	3641.44 (6130.1)	6.0

Table 1b. Experimental Results for 3TSP

3TSP	20 nodes	% impr.	40 nodes	% impr.	60 nodes	% impr.	80 nodes	% impr.
INIT	2500.17		3133.71		3488.88		4082.60	
INIT +2OPT	2338.31 (8.1)	6.5	2965.38 (19.5)	5.4	3338.86 (46.0)	4.3	3878.05 (88.2)	5.0
INIT +2OPT*	2283.96 (21.1)	8.6	2916.28 (55.7)	6.9	3286.53 (130.1)	5.8	3804.98 (265.5)	6.8
INIT +3OPT	2219.07 (190.8)	11.2	2837.22 (868.7)	9.5	3167.90 (2971.6)	9.2	3643.00 (7909.0)	10.8

This set of experiments shows the superiority of the 2-opt* exchange procedure over the classical 2-opt exchange procedure with respect to the total length of the routes. However, in about 60% of the problems, the two methods produced the same solution. It means that the general approach does not guarantee at all that a better solution will be found, but it is not costly to try it an it could provide great benefits. Generally speaking, the 2-opt* exchange procedure looks here as a good trade-off between low computation time and high solution quality, that is, a good compromise between the 2-opt and 3-opt exchange procedures.

5.2 MTSP With Time Windows

We then ran a new set of experiments to evaluate the general approach for MTSP problems with time windows. Due to the major increase in computation time relating to the explicit consideration of time windows during the exchange procedure, we restricted ourselves to networks with 20 and 40 nodes (20 networks for each size). Time windows were created by generating at random their lower and upper bounds. We generated feasible initial tours for two and three salesmen using a modified "farthest insertion" algorithm taking care of the temporal restrictions. Basically, we modified the algorithm so as to insert nodes at the minimal extra-mileage location for which time constraints were all satisfied. We then applied in turn the 2-opt, 2-opt* and 3-opt exchange procedures to these initial tours (only one improvement procedure was applied to each initial tour). Table II represents for each network size and for each algorithm, the total length of the final tours and the percentage of improvement over the initial solution (INIT). Computation times in seconds are shown within parentheses under each result, each one of them being the mean taken over twenty problems.

It must be noticed here that the initial solutions, as produced by our modified farthest insertion algorithm, were not as good as in the non-constrained case. This is not too surprising because the farthest insertion algorithm has proven to be successful for classical "constraint free" TSP problems. We simply modified this algorithm in a very straightforward manner so as to obtain initial feasible solutions taking into account the temporal restrictions. This explains the substantial improvement that has been achieved by the 2-opt* and 3-opt exchange procedures when applied to these initial solutions (as compared with the first set of experiments).

As we can see in Table II, the 2-opt* exchange procedure is much better than the classical 2-opt. Moreover, the 2-opt* now performs as well as a classical 3-opt. Once again, these impressive results stem from the fact that the general approach considers exchanges leading to new feasible solutions, which is not the case for a 2-opt exchange procedure. A close analysis of the 3-opt exchange procedure would show that only one exchange, from the seven possible exchanges leading to new solutions, has high probability of being feasible. The other alternatives are unlikely to be feasible (for the same reasons as those mentioned in section 4). The 2-opt* exchange procedure for the MTSP is thus very powerful when dealing with problems with time windows. In this context, it performs as well as a 3-opt exchange procedure, but the complexity of step 2 of the 2-opt* procedure (as described in section 3) remains in the order of n^2.

Table 2a. Experimental Results for 2TSP with time windows

2TSP	20 nodes	% impr.	40 nodes	% impr.
INIT	3812.60		4751.22	
INIT +2OPT	3758.11 (28.3)	1.5	4699.14 (112.0)	1.1
INIT +2OPT*	2983.54 (113.4)	21.8	3672.55 (457.8)	22.8
INIT +3OPT	3023.5 (970.2)	20.7	3660.81 (8111.6)	23.0

Table 2b. Experimental Results for 3TSP with time windows

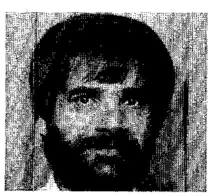
3TSP	20 nodes	% impr.	40 nodes	% impr.
INIT	4169.63		5011.27	
INIT +2OPT	4057.00 (32.3)	2.7	4920.98 (123.1)	1.8
INIT +2OPT*	2938.85 (140.7)	29.5	3633.18 (576.9)	27.5
INIT +3OPT	3004.22 (1030.4)	28.0	3658.22 (8243.9)	27.0

6. CONCLUSION

This paper has described a technique that generalizes the classical k-opt exchange procedure for the MTSP. Our experiments have shown that the technique is particularly powerful in the context of MTSP problems with time windows. In such a case, the generalized 2-opt exchange procedure is as powerful as a classical 3-opt exchange procedure, but shows a complexity that compares to the classical 2-opt exchange procedure.

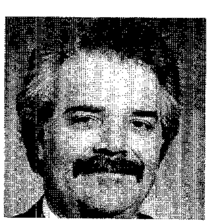
REFERENCES

1. Bellmore A. & Hong S., (1974), "Transformation of Multi-Salesman Problem to the Standard Traveling Salesman Problem", *J.ACM* 21, pp. 500-504.
2. Lapalme G., Potvin J.Y. & Rousseau J.M., (1988), "A General Heuristic for Node Routing Problems". in *Advances in Optimization and Control*, H.A Eiselt & G. Pederzoli (eds), Springer-Verlag, pp. 124-143.
3. Lin S., (1965), "Computer Solutions of the Traveling Salesman Problem", *Bell System Tech J.* 44, pp. 2245-2269.
4. Lin S. & Kernighan B., (1973), "An effective Heuristic Algorithm for the Traveling Salesman Problem", *Op. Res.* 21, pp. 498-516.
5. Rosenkrantz D., Sterns R & Lewis P., (1977), "An Analysis of several Heuristics for the Traveling Salesman Problem", *SIAM J Comp.* 6, pp. 563-581.
6. Syslo M.M., Deo N. & Kowalik J.S., (1983), *Discrete Optimization Algorithms*, Prentice-Hall Inc..
7. Svestka J. & Huckfeldt V., (1973), "Computational Experience with an *M*-Salesmen Traveling Salesman Algoritm", *Management Sc.* 19, pp. 790-799.



Jean-Yves Potvin is Associate Researcher at the Centre de Recherche sur les Transports of Montreal University. He earned a B.Sc. in Mathematics, an M.Sc. in Operations Research, a Ph.D. in Computer Science from Montreal University and completed a postdoctoral fellowship at Carnegie-Mellon University (CMU). His current research interests are in the application of Artificial Intelligence techniques for managing complex transportation problems.

Guy Lapalme is Full Professor in the Department of Computer Science and Operations Research of Montreal University. He received a Ph.D. degree in Computer Science from Montreal University. He works in the application of Artificial Intelligence techniques in the domains of transportation problems and macromolecule modeling. He is also interested in the use of functional and logic programming for natural language generation.



Jean-Marc Rousseau is Full Professor in the Department of Computer Science and Operations Research of Montreal University. He received a Ph.D. degree in Operations Research from M.I.T. He has published in international journals several papers on routing and scheduling problems, mainly in the transportation sector.