Muslic — Complete tool documentation

Assignment of origin-destination matrices on timetable-based multimodal networks — parallel version (November 2026)

1. Introduction

For each origin-destination (OD) pair of a matrix, Muslic computes the minimum generalised-cost path on a network that mixes timetabled arcs (public transport lines) and so-called individual arcs (walking, road). It then assigns the OD demand to that path (all-or-nothing assignment) and writes results per OD, per arc, per node, per service and per movement.

Two computation directions

Direction	Code in the matrix	Meaning of the time given	What the program looks for
Departure	d	Earliest departure time from the origin	The cheapest path, with the resulting arrival time
Arrival	a	Latest arrival time at the destination	The cheapest path, computed backwards from the destination

How it works

- Time is expressed in minutes since midnight. A service whose arrival time is lower than its departure time is considered to arrive the next day (1440 minutes are added).
- Services are attached to day-by-day calendars: a line may run on certain days only.
- Each arc carries a label (cost, time, travel time, and so on). The search uses cost buckets, in the spirit of Dial's algorithm.
- ODs are computed in parallel on all cores. Results are written in the order of the matrix and are identical whatever the number of threads.
- Consecutive ODs with the same origin, day and time (direction d) or the same destination, day and time (direction a) share a single shortest-path computation.

Optional features (described in sections 7 and 8): movement penalties and prohibitions (time, toll, cost), sub-paths of n arcs, and early/late weights.

2. Running the tool and files

2.1 Command line

```
Muslic <network> <matrix> <output_prefix> <parameters> [<penalties>]
```

Argument	Role	Required
network	Network file (nodes, arcs, services)	yes
matrix	File of the ODs to compute	yes
output_prefix	Start of the name of all result files, for example results/test1	yes
parameters	Computation parameter file	yes
penalties	File of movement penalties and sub-paths	no

With fewer than 4 arguments the program does nothing. If the network file, the matrix or the parameter file cannot be found, no computation is run. The penalties file is simply ignored if it is missing. File names written inside the parameter file are informative only: the command-line names replace them.

2.2 Overview of files

File	Direction	Role
network	input	Describes nodes, arcs, lines, services and calendars
matrix	input	Lists the ODs (origin, destination, demand, day, time, direction)
parameters	input	Weights, caps, algorithm and output choices
penalties	input	Prohibitions, times, tolls and costs of movements and sub-paths
prefix_od.txt	output	One line per OD: time, cost, itinerary
prefix_chemins.txt	output	The arcs of each path (if path output is enabled)
prefix_temps.txt	output	The arcs reached by each computation (depending on the time output mode)
prefix_noeuds.txt	output	The best label of each node for each computation

File	Direction	Role
prefix_isoles.txt	output	The arcs not reached (if enabled)
prefix_detour.txt	output	Detour report per origin (time output mode 0)
prefix_aff.txt	output	Cumulated volumes per arc
prefix_services.txt	output	Cumulated volumes per service (if enabled)
prefix_transferts.txt	output	Cumulated volumes per movement (if enabled)
prefix_param.txt	output	Copy of the parameters used
prefix_log.txt	output	Computation log

2.3 Execution

- During execution the console shows the progress of four steps: Network import, Network topology generation, Penalties and transfers import, then Shortest paths computing.
- All cores are used. The MUSLIC_THREADS environment variable sets the number of threads. Memory use grows with that number, because each thread works on its own copy of the arcs and services.
- Text files are UTF-8. Fields are separated by semicolons. When reading, both the period and the comma are accepted as the decimal separator. When writing, the separator follows the machine's regional settings.

3. Concepts

Notion	Definition
Node	Point of the network, identified by a text. It has x and y coordinates and an optional label
Arc	Directed section between two nodes, characterised by a line, a time, a length, a type and a toll
Line	Number of an arc. A strictly positive value means a public transport line with services. A negative or zero value means an individual arc (by convention -1)

Notion	Definition
Individual arc	Arc without timetable (walking, road). Its time is constant
Service	One timetabled run on a line arc: departure time, arrival time, calendar
Calendar	String of O (runs) or N (does not run), one per day. Identical strings form a regime
Type	Text attached to each arc (0 by default). It is used to apply different weights per arc category
Day	Index of the day in the calendars. It starts at 0
Label	Best known state of an arc: cost, arrival time, travel time, wait, number of boardings
Pole	Origin node of the trip, or the boarding node of the first vehicle

3.1 Generalised cost

The cost of a path is the sum of the costs of its arcs, plus the costs of transfers and penalties.

Element	Cost added
Individual arc	time × individual travel time factor × individual mode weight + arc toll × toll weight
Line arc	in-vehicle time × in-vehicle weight + wait × wait weight + arc toll × toll weight
Boarding with transfer	minimum transfer time × boarding weight
Movement penalties	time, toll × toll weight, and cost from the penalties file

Weights depend on the arc type (section 5.2). The actual trip time (columns temps, tveh, tmap, tatt, tcorr) is separate from cost: a weight of 2 doubles the cost of one minute without changing the arrival time.

3.2 Result quantities

Column	Meaning
heureo, heured	Requested time and arrival time of the path
temps	Arrival time minus requested time
tveh	Time spent in vehicles
tmap	Time spent on individual arcs, multiplied by their time factor
tatt	Total waiting time at stops
tcorr	Cumulated transfer time
ncorr	Number of boardings
tatt1	Wait at the first boarding
cout	Generalised cost
longueur	Cumulated length
pole	See the table of notions
volau	Assigned demand
toll	Cumulated tolls

4. Network file

The file has two blocks, introduced by a line containing t nodes then t links. Empty lines are ignored.

4.1 Node block

```
t nodes
identifier;x;y;label
```

Column	Content
1	Node identifier (text, without superfluous spaces)
2, 3	Coordinates x and y (numbers)
4	Label (optional)

A node that is already declared is ignored. A node cited in an arc without having been declared is created without coordinates.

4.2 Arc block

```
t links
ni;nj;time;length;line;service;hd;hf;calendar;text;type;toll
```

Column	Field	Content
1	ni	Start node
2	nj	End node
3	time	Time of the individual arc, in minutes (not used for the in-vehicle time of lines)
4	length	Arc length
5	line	Line number. Positive: public transport. Negative or zero: individual arc
6	service	Service number. Strictly positive for a service; -1 for an individual arc
7	hd	Departure time of the service, in minutes since midnight
8	hf	Arrival time of the service. If hf is lower than hd, 1440 is added to it
9	calendar	String of O and N, one character per day (for example OOOONNO)
10	text	Arc label, for example the line name. The first element separated by
11	type	Arc type (0 by default)
12	toll	Arc toll (0 by default)

Rules

- An arc is identified by the triplet start node, end node, line. Several lines of the file with the same triplet and a positive service number add services to the same arc. All the other columns are those of the first line.
- An individual arc is written on a single line, with service number -1.
- The types used in the network determine the keys of the per-type weights in the parameter file.

Example

```
t nodes
1;0;0;north station
2;100;0;south station
3;200;0;centre
t links
1;2;10;1000;1;1;395;405;OOOO;Line A;1;0
1;2;10;1000;1;2;410;420;OOOO;Line A;1;0
2;3;4;300;-1;-1;0;0;OOOO;walk;0;0
```

The first two arc lines describe two services of line 1 on arc 1→2 (departures at 395 and 410). The last one describes a walking arc.

5. Parameter file

One value per line. The text after the first semicolon is a label that is ignored when reading: the _param.txt file written by the program uses the same format. The first 21 lines are mandatory, in order. The following ones are optional, but in practice they must be filled in: the default value of the maximum individual travel time is 0, which forbids individual arcs.

5.1 File lines

No.	Usual label	Type	Role
1	algorithm	integer	0: processing in arrival order inside a bucket (fast, approximate). 1: exact minimum in the bucket (default). 2: not implemented
2	prohibited U-turns	boolean	True: forbids immediately going back onto the reverse arc (same node)
3	max buckets	number	Maximum number of buckets of the computation queue (10000 by default)
4	number of days	integer	Number of extra days explored to find services after the departure day
5	matrix file	text	Informative: replaced by the command line
6	turns and transfers file	text	Informative: replaced by the command line

No.	Usual label	Type	Role
7	network file	text	Informative: replaced by the command line
8	generic output file	text	Informative: replaced by the command line
9	algorithm parameter	integer	Bucket scale: cost represented by one bucket
10	algorithm power	number	Exponent of the bucket indexing function
11	output paths	boolean	Writes the _chemins file
12	output services	boolean	Writes the _services file
13	output travel times	integer	Time output mode: 0, 1, 2 or 3 (see 5.3). Add 10 to enable the stops column
14	output turns and transfers	boolean	Writes the _transferts file
15	boarding weight	weight per type	Boarding weight
16	individual mode weight	weight per type	Weight of time on individual arcs
17	individual travel time factor	weight per type	Multiplying factor of the time of individual arcs (speed)
18	in-vehicle time weight	weight per type	Weight of in-vehicle time
19	wait time weight	weight per type	Weight of waiting time
20	min transfer time	duration per type	Minimum time between an arrival and a boarding, in minutes
21	max transfer time	duration per type	Maximum wait accepted before a boarding, in minutes
22	max individual travel time	number	Cap on the individual time cumulated along a path, in minutes
23	toll weight	weight per type	Conversion of toll into cost

No.	Usual label	Type	Role
24	output filter types	list	Arc types kept in the outputs, separated by
25	max travel cost	number	Maximum generalised cost (120 by default). It bounds the search and the outputs
26	output nodes	boolean	Writes the _noeuds file (True by default)
27	output isolated links	boolean	Writes the _isoles file (False by default)
28	early departure/arrival weight	number	Early weight (optional, see section 7.4)
29	late departure/arrival weight	number	Late weight (optional, see section 7.4)

5.2 Values per arc type

Lines 15 to 21 and 23 accept one value per arc type, with the syntax default_value|type:value|type:value.

```
1|v:1|w:1.1|x:1.05;individual travel time factor
```

Here the factor is 1 by default, 1.1 for arcs of type w and 1.05 for those of type x. The types of the line are separated by the character |, and each type is separated from its value by a colon. The semicolon only introduces the label: a value written after a semicolon would be ignored. A type that is absent from the line takes the default value. A line without types is simply written 1.

5.3 Time output modes (line 13)

Value	Effect
0	No _temps file. Writes the _detour file (report per origin)
1	_temps contains the individual arcs reached
2	_temps also contains the line arcs reached. Line arcs also count for nodes

Value	Effect
3	Compact format (origin, arc, time, wait at the first boarding, volume). If node output is enabled, only _noeuds is written
+10	Adds to _noeuds the column of stops used (stops)

5.4 Example file

```
1;algorithm
True;prohibited U-turns
10000;max buckets
2;number of days
network.txt;matrix file
pen.txt;turns and transfers file
network.txt;network file
res;generic output file
1;algorithm parameter
1;algorithm power
True;output paths
True;output services
2;output travel times
True;output turns and transfers
5;boarding weight
1;individual mode weight
1;individual travel time factor
1;in-vehicle time weight
1;wait time weight
2;min transfer time
30;max transfer time
60;max individual travel time
0;toll weight
;output filter types
150;max travel cost
True;output nodes
False;output isolated links
```

6. OD matrix file

One line per OD, fields separated by semicolons. Empty lines are ignored.

6.1 Basic format

```
origin;destination;demand;day;time;direction;label
```

Column	Content
1	Identifier of the origin node
2	Identifier of the destination node
3	Demand (volume) assigned to the path
4	Day: index in the calendars, starting at 0
5	Time, in minutes since midnight
6	Direction: d (earliest departure) or a (latest arrival). Optional: by default, direction d
7	OD label, reused in the id column of the results. Optional: by default, the line's sequence number

If column 6 contains anything other than d or a, the direction of the previous line is kept.

Examples

```
1300;1600;2.5;0;400;d;trip1
1300;1600;1;0;401;a;trip2
1100;1800;1;0;420
```

The third line has no direction and no label: it is treated as departure direction.

6.2 Line order and reuse of computations

To avoid useless computations, sort the matrix by origin, day and departure time for direction d lines, and by destination, day and arrival time for direction a lines. A line reuses the shortest-path tree of the previous line if all three conditions hold:

- same origin, day, time and direction (same destination in direction a);
- fewer than 13 columns;
- parameters identical to those of the previous line.

For a reused OD, the nbpop column of the result is 0.

6.3 Advanced mode: parameters specific to a line

A line can carry its own parameters in the following columns. Columns are positional and empty columns are dropped when reading: fill in all columns up to the last one used.

Columns	Content	Read if the line has
8	In-vehicle weight	at least 18 columns
9	Wait weight	same
10	Individual mode weight	same
11	Boarding weight	same
12	Individual travel time factor	same
13	Minimum transfer time	same
14	Maximum transfer time	same
15	Number of days	same
16	Maximum individual time	same
17	Toll weight	same
18	Maximum cost	same
19	Path output (True or False)	at least 24 columns
20	Time output mode	same
21	Algorithm	same
22	Algorithm parameter	same
23	Maximum number of buckets	same
24	Power	same
25	Output types filter	at least 25 columns
26	Early weight	at least 26 columns
27	Late weight	at least 27 columns

Columns 8, 9, 10, 11, 12, 13, 14 and 17 accept the per-type syntax of section 5.2.

Rules

- A line without these columns uses the parameters of the parameter file. The values of one line do not carry over to the following ones.

- A default value given alone (for example 5 for the wait weight) applies to all arc types.
- The filter in column 25 must be filled in for columns 26 and 27 to be read. To avoid filtering, list all types separated by | .
- The parameters of a line are not copied to _param.txt or _log.txt.

Example (wait weight 5, with other coefficients taken from the example values)

```
1300;1600;1;0;400;d;ex;1;5;1;5;1;2;30;2;60;0;150;True;2;1;1;10000;1;0|1;0.5;1
```

7. Penalties file

This file is optional. It describes movements (passage from one arc to the next) and sequences of arcs to forbid or penalise. It accepts two kinds of lines.

7.1 Movement lines (classic format)

```
via_node;from_node;from_line;to_node;to_line;time;toll
```

Field	Meaning
via_node	Node where the movement takes place
from_node, from_line	Start node and line of the arc arriving at the via node
to_node, to_line	End node and line of the arc leaving the via node
time	Movement time in minutes. Negative value: movement forbidden. 0: no time penalty
toll	Optional. Toll of the movement, positive or negative

Effects

- Time: for a transfer between two lines, the time is added to the minimum transfer time, in both computation directions. For a movement involving an individual arc, it is added to the travel time.
- Toll: multiplied by the toll weight of the type of the arc taken after the movement, it is added to the cost. The toll itself is added to the toll column. A negative toll acts as a bonus, capped at the cost of the previous arc.
- Prohibition: a negative time forbids the movement and makes the toll irrelevant.

A line with an unknown node or arc is ignored. If the same movement is defined several times, the first definition is kept. The nodes involved are automatically flagged as intersections, and volumes per movement are written to _transferts.txt if that output is enabled.

7.2 P lines: sub-paths and cost

```
P;time;cost;toll;node0;line1;node1;line2;node2;line3;node3...
```

A P line describes a sequence of at least two arcs. Each arc is given by its start node, its line and its end node.

Field	Meaning
P	Marks the line
time	Time in minutes. Negative value: sequence forbidden
cost	Penalty cost, positive or negative, in generalised cost units
toll	Penalty toll, positive or negative
node0, line1, node1, ...	The sequence of arcs: the end node of one arc is the start node of the next

Rules

- With 2 arcs, the P line acts like a movement line with an additional cost.
- With 3 arcs or more, the penalty applies when the sequence is complete. A path that follows only the beginning of it is not penalised.
- For each arc the program keeps the best path that falls into no forbidden sequence: an arc that is reachable by another path stays reachable.
- If several sequences end at the same time: the largest time, costs and tolls added, prohibition prevails.
- A line with an unknown node or arc is ignored and reported in the log.
- The bonus of a negative cost or toll is capped at the cost of the previous arc.

Examples (walking arcs, line -1)

```
P;-1;0;0;1;-1;2;-1;3;-1;4          forbids the sequence 1→2, 2→3, 3→4
P;0;10;0;1;-1;2;-1;3;-1;4          cost of 10 for this sequence
P;0;0;1.5;1;-1;2;-1;3;-1;4         toll of 1.5 for this sequence
P;0;5;0;2;-1;3;-1;4                cost of 5 for the movement 2→3, 3→4
```

7.3 Arc accessibility with prohibitions

An arc that is accessible without the prohibitions stays accessible if a path exists that uses no forbidden sequence. It becomes inaccessible only if all its paths use one. If the destination of an OD becomes inaccessible, no line is written for it. The caps on cost, individual time and transfer remain active: a replacement path that is too long can fall outside the search perimeter.

7.4 Early and late weights

These two weights allow leaving earlier (departure mode) or arriving later (arrival mode) than the requested time, at a cost per minute of deviation.

Coding in the parameter file: two lines at the end, after output isolated links.

```
0.5;early departure/arrival weight
1.0;late departure/arrival weight
```

Coding in the matrix: columns 26 (early) and 27 (late).

Principle

- Departure mode: let R be the arrival time at the boarding stop (requested time plus access walk) plus the minimum transfer time. The normal option is the first service departing at R or later: cost = trip + wait × wait weight + late weight × (departure − R). The early option is the last service departing before R, at most (max transfer time) before: cost = trip + early weight × (R − departure), with no waiting cost.
- Arrival mode: normal option = last service arriving at R or earlier (cost = trip + wait × weight + early weight × (R − arrival)). Late option = first service arriving after R (cost = trip + late weight × (arrival − R)).
- The alternative applies only to the first boarding (departure mode) and the last alighting (arrival mode), including after an access or egress walk, not to transfers.

Reading: tatt and tatt1 become negative when the alternative is chosen. A value of -5 in departure mode means leaving 5 minutes earlier than planned; in arrival mode, arriving 5 minutes late.

8. Result files

All files are written from the output prefix, with the suffix shown. They are created on every run: a file whose option is disabled contains only its header.

8.1 _od.txt: one line per OD

Column	Content
id	OD label
o, d	Origin and destination
jour	Day
heureo	Requested time
heured	Arrival time of the path
temps	heured minus heureo
tveh, tmap, tatt, tcorr	In-vehicle, individual-arc, waiting and transfer times
ncorr	Number of boardings
tatt1	Wait at the first boarding
cout	Generalised cost
longueur	Path length
pole	Origin or boarding node of the first vehicle
volau	Assigned demand
texte	Itinerary: labels of the successive arcs, separated by
nbpop	Number of labels processed by the computation (0 if the computation is reused)
toll	Cumulated tolls

An OD whose destination is not reachable has no line.

8.2 _chemins.txt: arcs of each path

Written if path output is enabled. One line per arc of the path, in order from the destination to the origin in direction d, from the origin to the destination in direction a.

Column	Content
id, o, d, jour, heure	OD and requested time
i, j, ij	Start and end nodes of the arc, and their concatenation
ligne, service	Line and service number used
temps, heureo	Time at the end of the arc minus the requested time, and time at the end of the arc
tveh, tmap, tatt, tcorr, ncorr, tatt1, cout, longueur, pole	Values cumulated up to this arc
volau, boai, alij	Demand, and demand that boards or alights on this arc
texte, type, toll	Label, type and toll of the arc

8.3 _temps.txt: arcs reached by each computation

Content depends on the time output mode (section 5.3). For modes 1 and 2, one line per arc reached, whose cost is below the maximum cost:

Column	Content
id, o	Computation (OD label) and origin
ij, ligne, numtrc	Arc (start node-end node), line and service number
jour, heureo, heured	Day, requested time and time at the end of the arc
temps, tveh, tmap, tatt, tcorr, ncorr, tatt1, cout, longueur, pole	Values cumulated up to this arc
volau	Assigned demand
precedent	Previous arc in the path
type, toll	Arc type and toll
ti	Time of entry onto the arc

Mode 3 writes compact lines: o;ij;ligne;temps;tatt1;volau.

8.4 _noeuds.txt: label of each node

Written if node output is enabled. One line per node reached: the best arriving arc at that node, whose cost does not exceed the maximum cost and whose type passes the filter. Line arcs are taken into account only if the time output mode is 2.

Column	Content
id, o, d, jour	Computation and day
numero	Node
heureo, heured, temps, tveh, tmap, tatt, tcorr, ncorr, tatt1, cout, longueur, pole, toll, volau	Same quantities as in _od, for the best path to the node
stops	Sequence of stops used (time output mode + 10)

With mode 3, the lines are compact: o;numero;temps;tatt1;volau.

8.5 Other files

File	Content
_isoles.txt	Written if enabled. One line per arc not reached: start node-end node; line; arc index. No header
_detour.txt	Written for time output mode 0. One line per origin: origin; sum of arrival times at the arcs reached; sum of straight-line distances to those arcs; number of arcs. Allows an average detour index to be computed. No header
_aff.txt	Cumulated volumes per arc: i;j;ligne;volau;boai;alij;texte;type;toll. volau is the volume crossing the arc, boai the one boarding and alij the one alighting
_services.txt	If enabled. Volumes per service: i;j;ligne;service;hd;hf;regime;volau;boia;alij;texte;type. boia is the volume boarding, alij the one alighting
_transferts.txt	If enabled. Volumes per movement: j;i;lignei;k;lignek;textei;textek;volau, with j the node of the movement

File	Content
_param.txt	Copy of the parameters used, in the format of the parameter file
_log.txt	Log: file names, parameters, start and end times, duration, anomalies (section 11)