Collaborative Exploratory Research: The Anticipatory Route Guidance Problem

SPR ID# C-05-04

FINAL REPORT

July, 2006

Submitted by

Soulaymane Kachani, Assistant Professor, Department of Industrial Engineering and Operations Research, Columbia University, 334 S.W. Mudd Building, 500 W. 120th Street, New York NY, 10027, Tel: 212-854-1804; Fax: 212-854-8103; Email: kachani@ieor.columbia.edu

Georgia Perakis, Associate Professor, MIT Operations Research Center, E53-359, 77 Massachusetts Avenue, Cambridge MA, 02139 Tel: 617-253-8277; Fax: 617-258-7579; Email: georgiap@mit.edu

The 2005 UTRC Research Initiative

Region II

In cooperation with

New York State Department of Transportation

and

U.S. Department of Transportation

Disclaimer Statement

TECHNICAL REPORT STANDARD TITLE PAGE

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle Collaborative Exploratory Research: The Anticipatory Route Guidance Problem		5. Report Date 6. Performing Organization Code
7. Author(s) Soulaymane Kachani, Assistant Professor, Columbia University, Georgia Perakis, Associate Professor, MIT Operations Research		8. Performing Organization Report No. 49777-1805
9. Performing Organization Name and Address Columbia University, 334 S.W. Mudd Building, 500 W. 120th Street, New York NY, 10027 Region 2- University Transportation Research Center, New York, NY 10031		10. Work Unit No. 11. Contract or Grant No.
12. Sponsoring Agency Name and Address New York State Department of Transportation 50 Wolf Road Albany, NY 12232		13. Type of Report and Period Covered Final Report July 1, 2005 - June 30, 2006 14. Sponsoring Agency Code
15. Supplementary Notes		

16. Abstract

Finding solutions to fixed point problems can help government and industry leaders to plan for real world success. One concrete example of problem solving which may be amenable to fixed point solution is the anticipatory route guidance problem (ARG). An exercise in dynamic traffic user-equilibrium, this problem envisions a communications system which transmits dynamic, shortest path traffic data to drivers. But anything that influences the path-choice decisions of drivers will, in itself, affect traffic conditions on the ground.

The challenge is clear: develop a model in which shortest-path forecasting does not become a self-defeating prophecy.

This research develops and evaluates a software system which explores the ARG problem from a fixed point perspective. A significant part of our research consists in identifying the best algorithms for step size computation. Methods evaluated include: MSA (Method of successive averaging), Polyak iterate averaging method, and a variety of potential optimization line search methods.

17. Key Words	18. Distribution Statement
Route guidance, Dynamic traffic user-equilibrium, Advanced traveler information system (ATIS), Traveler information	

19. Security Classif (of this report)	20. Security Classif. (of this page)	21. No of Pages	22. Price
Unclassified	Unclassified	104	

Form DOT F 1700.7 (8-69)

Table of Contents

Introduction

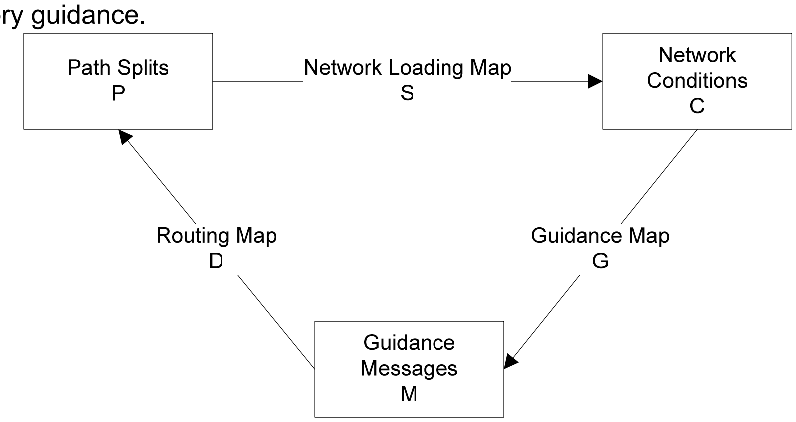
Finding solutions to fixed point problems can help government and industry leaders to plan for real world success. One concrete example of problem solving which may be amenable to fixed point solution is the anticipatory route guidance problem (ARG). An exercise in dynamic traffic user-equilibrium, this problem envisions a communications system which transmits dynamic, shortest path traffic data to drivers. But anything that influences the path-choice decisions of drivers will, in itself, affect traffic conditions on the ground.

The challenge is clear: develop a model in which shortest-path forecasting does not become a self-defeating prophecy.

This research develops and evaluates a software system which explores the ARG problem from a fixed point perspective. A significant part of our research consists in identifying the best algorithms for step size computation. Methods evaluated include: MSA (Method of successive averaging), Polyak iterate averaging method, and a variety of potential optimization line search methods.

Consistent Anticipatory Route Guidance

A basic framework for analyzing anticipatory route guidance is defined by three time-dependent variables and three maps that interrelate them. The variables of interest are path splits, network conditions and guidance messages. The maps of interest are the network loading map, the guidance map and the routing map. These maps can be combined in a variety of ways to obtain composite relationships needed to define and compute consistent anticipatory guidance.



The above lead naturally to the definition of composite maps that combine the dynamic network loading, routing and guidance maps in different sequences. This research dealt with 2 out of 3 possible composite maps:

- a composite map $D \circ G \circ S : P \mapsto P$ from the domain path splits into itself, which starts with a set of time-dependent path splits, forecasts the corresponding network conditions, determines an appropriate set of guidance messages, which are disseminated to drivers and cause them to react in some way, leading to a new set of path splits.
- a composite map $S \circ D \circ G : C \mapsto C$ from the domain of network conditions into itself. This map begins with a set of time-varying network conditions and determines the messages with which the guidance system responds; these are communicated to drivers and affect the path splits, thus resulting in a new set of network conditions.

Evaluating one of these composite framework maps corresponds to executing a one-pass forecasting model that invokes the component maps in the indicated order of composition. Input to the model is an assumption about the time trajectory of one of the framework variables (splits, conditions or messages); its output is a prediction of a possibly different trajectory of the same variable. This can be written as:

$$model(assumption) = predictions$$

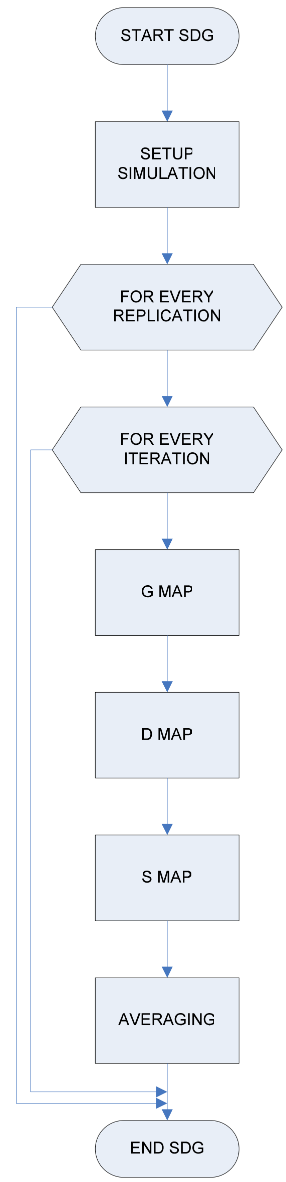
The guidance generated by a model is said to be consistent when the assumptions used as the basis for generating it prove to be verified, within the logic of the predictive model, after drivers receive the guidance and react to it. In terms of the composite framework maps, consistency means that a map's predicted time-dependent outputs coincide with its assumed time-dependent inputs. Again, this can be written as:

$$model(assumption) = predictions = assumptions$$

For the composite path split map, guidance is consistent if the forecast path splits coincide with the splits that were assumed at the start. For the composite network condition map, guidance is consistent if the initial network conditions used for the guidance determination coincide with those that result after the guidance is disseminated. The coincidence of the assumed input value with the corresponding predicted output value of a composite framework map can be expressed by saying that the value is a fixed point of the map.

Computational Implementation and Tests

An application was developed as part of this research to operationalize and explore properties of the route guidance fixed point framework and to investigate the performance of various algorithmic approaches. The system is coded in C++. The following diagram shows the implemented algorithm in the code at a glance, particularly for a composite map $S \circ D \circ G : C \mapsto C$.



Network Loading Map (S)

The application is a traffic simulator. The network loading map is a discrete-time vehicle based traffic simulator that implements a store-and-forward protocol with blocking (spillback). The loader's inputs are the network description in terms of nodes, links and paths. The loader's output is a table of average link traversal times by link and by time of link entry.

It represents links as deterministic FIFO (first-in-first-out) single-server queues with given exit (service) and storage capacities. Link attributes include connectivity (and node numbers), length, fixed speed, number of lanes and per lane exit capacity (Bottom, 2000).

Guidance Map (G)

The application can represent descriptive and prescriptive guidance.

Descriptive guidance is implemented as tables of time-dependent path times from decision points to destinations, with separate tables maintained for ubiquitous and short-range guidance. For the prescriptive guidance, the guidance map simply designates a path having minimum travel time from the guidance link to each possible destination, using the latest estimates of time-dependent link traversal times for the path time comparisons (Bottom, 2000).

Routing Map (D)

The simulator's routing map is based on an underlying model of driver route choice behavior that predicts path choice probabilities. These become path splits in one of two ways, at the user's option. In an aggregate application of the probabilities, they are assumed to apply to homogeneous group of vehicles as a whole. In this case, the path splits are identical to the choice probabilities. In a disaggregate application, each individual vehicle is allocated a path based on an independent draw form the choice probabilities.

Algorithms

This section explains in more detail about parts of the algorithm that were studied particularly more closely during the research.

Averaging

Constant Step Size Averaging Algorithm

Algorithm

The algorithm computes the conditions C^{i+1} in iteration $i+1$ in terms of those in iteration i as

$$C^{i+1} = C^{i} + \alpha\left(S \circ D \circ G\left(C^{i}\right) - C^{i}\right);\ i = 0\ldots$$

Code Implementation

- Take the value `Stepsize computation parameter` in `Run Control File` (for example, `SDG-test.run`) and assign it to variable `alpha`.
- Store C^{i} to `ltt0`, in case of SDG mapping.
- Simulate the mapping and store $S \circ D \circ G\left(C^{i}\right)$ to `ltt1`, in case of SDG mapping.
- Call `average_link_time_tables`, in case of SDG mapping (or `average_path_split_tables`, in case of DGS mapping, or `average_msgs`, in case of GSD mapping) in order to get C^{i+1}.
-
```
void average_link_time_tables (float alpha, link_time_table * ltt1, link_time_table * ltt2)
{
  if (! (ltt1->granularity == ltt2->granularity && ltt1->nslices == ltt2->nslices)) {
    cerr << "Attempt to average incommensurate link time tables !" << endl;
    exit (1);
  }
  int nl = nw->get_nlinks (), ns = ltt1->nslices;
  for (int il = 0; il < nl; il++) //mw2221 for each links...
    for (int ip = 0; ip < ns; ip++) //mw2221 for each period...
      ltt1->lttab[il][ip] //mw2221 calculate the average and populate link time table ltt1
        = (PERIOD) ((1.0-alpha) * abs(ltt1->lttab[il][ip])
                   + alpha * abs(ltt2->lttab[il][ip]) + 0.5);
}
```
-
```
void average_path_split_tables
  (float alpha, path_split_table * pst1, path_split_table * pst2)
{
  const float calpha = 1. - alpha;
  int nz = nw->get_nzones (), nn = nw->get_nnodes ();

  for (int io = 0; io < nz; io++)
    if (pst1->pretriptab[io]) {
      for (int ic = 0; ic < 2; ic++)
        for (int id = 0; id < nz; id++)
          for (int ip = 0, np = ph->get_npaths (io, id); ip < np; ip++)
            for (int it = 0; it < ninflowperiods; it++)
              pst1->pretriptab[io][ic][id][ip][it] =
                calpha  * pst1->pretriptab [io][ic][id][ip][it]
                + alpha * pst2->pretriptab [io][ic][id][ip][it];

    }

  for (int in = 0; in < nn; in++)
    if (pst1->enroutetab[in]) {
      for (int id = 0; id < nz; id++)
        for (int fp = 0, np = ph->get_npaths (in, id); fp < np; fp++)
          for (int tp = 0; tp < np; tp++)
            for (int it = 0; it < nperiods; it++)
              pst1->enroutetab[in][id][fp][tp][it] =
                calpha  * pst1->enroutetab[in][id][fp][tp][it]
                + alpha * pst2->enroutetab[in][id][fp][tp][it];
    }
}
```

Dataset Modeling

In order to activate this type of averaging, set `Stepsize computation option` in `Run Control File` (for example, `SDG-test.run`) to `0`. Also, specify the `Stepsize computation parameter` in `Run Control File`.

MSA (Method of Successive Averaging) Algorithm

Algorithm

The following is the description of the algorithm (Bottom, 2000, page 130):

> The MSA fixed point solution algorithm computes the conditions C^{i+1} in iteration $i+1$ in terms of those in iteration i as
>
> $$C^{i+1} = C^{i} + \frac{1}{i+1}\left(S \circ D \circ G\left(C^{i}\right) - C^{i}\right);\ i = 0\ldots$$
>
> The correction $S \circ D \circ G\left(C^{i}\right) - C^{i}$ that the MSA applies to iteration i 's estimate C^{i} of the fixed point solution is weighted by $1/(i+1)$, with the overall correction becoming negligible as some point.

Code Implementation

- Calculate $\frac{1}{i+1}$ and assign it to variable `alpha`.
- Store C^{i} to `ltt0`, in case of SDG mapping.
- Simulate the mapping and store $S \circ D \circ G\left(C^{i}\right)$ to `ltt1`, in case of SDG mapping.
- Call `average_link_time_tables`, in case of SDG mapping (or `average_path_split_tables`, in case of DGS mapping, or `average_msgs`, in case of GSD mapping) in order to get C^{i+1}.
  -
```
void average_link_time_tables (float alpha, link_time_table * ltt1, link_time_table * ltt2)
{
  if (! (ltt1->granularity == ltt2->granularity && ltt1->nslices == ltt2->nslices)) {
    cerr << "Attempt to average incommensurate link time tables !" << endl;
    exit (1);
  }
  int nl = nw->get_nlinks (), ns = ltt1->nslices;
  for (int il = 0; il < nl; il++) //mw2221 for each links...
    for (int ip = 0; ip < ns; ip++) //mw2221 for each period...
      ltt1->lttab[il][ip] //mw2221 calculate the average and populate link time table ltt1
        = (PERIOD) ((1.0-alpha) * abs(ltt1->lttab[il][ip])
                   + alpha * abs(ltt2->lttab[il][ip]) + 0.5);
}
```
  -
```
void average_path_split_tables
  (float alpha, path_split_table * pst1, path_split_table * pst2)
{
  const float calpha = 1. - alpha;
  int nz = nw->get_nzones (), nn = nw->get_nnodes ();

  for (int io = 0; io < nz; io++)
    if (pst1->pretriptab[io]) {
      for (int ic = 0; ic < 2; ic++)
        for (int id = 0; id < nz; id++)
          for (int ip = 0, np = ph->get_npaths (io, id); ip < np; ip++)
            for (int it = 0; it < ninflowperiods; it++)
              pst1->pretriptab[io][ic][id][ip][it] =
                calpha  * pst1->pretriptab [io][ic][id][ip][it]
                + alpha * pst2->pretriptab [io][ic][id][ip][it];

    }

  for (int in = 0; in < nn; in++)
    if (pst1->enroutetab[in]) {
      for (int id = 0; id < nz; id++)
        for (int fp = 0, np = ph->get_npaths (in, id); fp < np; fp++)
          for (int tp = 0; tp < np; tp++)
            for (int it = 0; it < nperiods; it++)
              pst1->enroutetab[in][id][fp][tp][it] =
                calpha  * pst1->enroutetab[in][id][fp][tp][it]
                + alpha * pst2->enroutetab[in][id][fp][tp][it];
    }
}
```

Dataset Modeling

In order to activate this type of averaging, set `Stepsize computation option` in `Run Control File` (for example, `SDG-test.run`) to `1`.

Polyak Iterate Averaging Algorithm

Algorithm

The algorithm is described as follows (Bottom, 2000, page 136):

When applied to find a fixed point of the $S \circ D \circ G$ map, the Polyak algorithm can be written:

$$C^{i+1} = C^i + \alpha^i \left(S \circ D \circ G\left(C^i\right) - C^i \right);\ i = 0\ldots$$

$$\tilde{C}^{i+1} = \tilde{C}^i + \frac{1}{i+1} C^{i+1}$$

where $\tilde{C}^n$ in the final iteration n is the fixed point estimate.

In practice, iterate averaging (computation of the $\tilde{C}^i$ s) is only started after the MSA-like step shows signs of stabilizing. (This is called the *window of averaging*.)

Step sizes α^i are frequently generated by formula such as $\beta i^{-\gamma}$, with $\gamma \in (1/2, 1)$. A common choice is $\gamma = 2/3$. In all runs presented here, $\beta = 1$ and $\gamma = 2/3$.

Code Implementation

- Take the value `Stepsize computation parameter` in `Run Control File` (for example, `SDG-test.run`), multiply it by $i^{2/3}$ and assign the result to variable `alpha`.
- Store C^i to `ltt0`, in case of SDG mapping.
- Simulate the mapping and store $S \circ D \circ G\left(C^i\right)$ to `ltt1`, in case of SDG mapping.
- Call `average_link_time_tables`, in case of SDG mapping (or `average_path_split_tables`, in case of DGS mapping, or `average_msgs`, in case of GSD mapping) in order to get C^{i+1}.
  - ```
    void average_link_time_tables (float alpha, link_time_table * ltt1, link_time_table * ltt2)
    {
      if (! (ltt1->granularity == ltt2->granularity && ltt1->nslices == ltt2->nslices)) {
        cerr << "Attempt to average incommensurate link time tables !" << endl;
        exit (1);
      }
      int nl = nw->get_nlinks (), ns = ltt1->nslices;
      for (int il = 0; il < nl; il++) //mw2221 for each links...
        for (int ip = 0; ip < ns; ip++) //mw2221 for each period...
          ltt1->lttab[il][ip] //mw2221 calculate the average and populate link time table ltt1
            = (PERIOD) ((1.0-alpha) * abs(ltt1->lttab[il][ip])
                       + alpha * abs(ltt2->lttab[il][ip]) + 0.5);
    }
    ```
  - ```
    void average_path_split_tables
      (float alpha, path_split_table * pst1, path_split_table * pst2)
    {
      const float calpha = 1. - alpha;
      int nz = nw->get_nzones (), nn = nw->get_nnodes ();

      for (int io = 0; io < nz; io++)
        if (pst1->pretriptab[io]) {
          for (int ic = 0; ic < 2; ic++)
            for (int id = 0; id < nz; id++)
              for (int ip = 0, np = ph->get_npaths (io, id); ip < np; ip++)
                for (int it = 0; it < ninflowperiods; it++)
                  pst1->pretriptab[io][ic][id][ip][it] =
                    calpha  * pst1->pretriptab [io][ic][id][ip][it]
                    + alpha * pst2->pretriptab [io][ic][id][ip][it];

        }

      for (int in = 0; in < nn; in++)
        if (pst1->enroutetab[in]) {
          for (int id = 0; id < nz; id++)
            for (int fp = 0, np = ph->get_npaths (in, id); fp < np; fp++)
              for (int tp = 0; tp < np; tp++)
                for (int it = 0; it < nperiods; it++)
                  pst1->enroutetab[in][id][fp][tp][it] =
                    calpha  * pst1->enroutetab[in][id][fp][tp][it]
                    + alpha * pst2->enroutetab[in][id][fp][tp][it];
        }
    }
    ```
- After the first set of iterations is done, take C^j. j is the iteration at which the MSA-like step shows signs of stabilizing. It is "arbitrarily" defined by start_iter. Store C^j to `ltt1`, in case of SDG mapping. Initiate `ltt0`, in case of SDG mapping. The next set of iterations starts at start_iter.
  - Calculate $\dfrac{1}{i - j + 1}$ and assign it to variable `alpha`.
  - Call `average_link_time_tables`, in case of SDG mapping (or `average_path_split_tables`, in case of DGS mapping, or `average_msgs`, in case of GSD mapping) in order to get C^{i+1}. This will update `ltt0`.
  - Simulate the mapping and store $S \circ D \circ G\left(C^i\right)$ to `ltt1`, in case of SDG mapping.

Dataset Modeling

In order to activate this type of averaging, set `Stepsize computation option` in `Run Control File` (for example, `SDG-test.run`) to 5. Also, specify the `Stepsize computation parameter` in `Run Control File`.

Potential Optimization Line Search Algorithm

Algorithm

The algorithm can be summarized as follows:

When applied to find a fixed point of the $S \circ D \circ G$ map, the potential optimization line search algorithm can be written:

$$C(\alpha) = C^{i+1} = C^i + \alpha^i \left(S \circ D \circ G\left(C^i\right) - C^i\right);\ i = 0\ldots$$

For each iteration, α is chosen by minimizing different potentials. In this research, the following minimization was investigated:

$$\min_{\alpha} \left\| S \circ D \circ G\left(C^i(\alpha)\right) - C^i(\alpha) \right\|^2 - \beta\alpha(1-\alpha)\left\| S \circ D \circ G\left(C^i\right) - C^i \right\|^2$$

$$\min_{\alpha} \left\| S \circ D \circ G\left(C^i(\alpha)\right) - C^i(\alpha) \right\|^2 - \beta\alpha^2\left\| S \circ D \circ G\left(C^i\right) - C^i \right\|^2$$

$$\min_{\alpha} \left\| S \circ D \circ G\left(C^i(\alpha)\right) - C^i(\alpha) \right\| - \beta\alpha(1-\alpha)\left\| S \circ D \circ G\left(C^i\right) - C^i \right\|$$

$$\min_{\alpha} \ln\left(\left\| S \circ D \circ G\left(C^i(\alpha)\right) - C^i(\alpha) \right\|^2 \right) - \beta \ln\left(\alpha(1-\alpha)\left\| S \circ D \circ G\left(C^i\right) - C^i \right\|^2 \right)$$

$$\min_{\alpha} \ln\left(\left\| S \circ D \circ G\left(C^i(\alpha)\right) - C^i(\alpha) \right\|^2 \right) - \beta \ln\left(\alpha\left\| S \circ D \circ G\left(C^i\right) - C^i \right\| \right) \ln\left((1-\alpha)\left\| S \circ D \circ G\left(C^i\right) - C^i \right\| \right)$$

$$\min_{\alpha} \left(\left(S \circ D \circ G\left(C^i\right) - C^i\right)^t \left(S \circ D \circ G\left(C^i(\alpha)\right) - C^i(\alpha)\right) \right)^2 - \beta\alpha(1-\alpha)\left\| S \circ D \circ G\left(C^i\right) - C^i \right\|^4$$

$$\min_{\alpha} \left(\left(S \circ D \circ G\left(C^i\right) - C^i\right)^t \left(S \circ D \circ G\left(C^i(\alpha)\right) - C^i(\alpha)\right) \right)^2 - \beta\alpha^2\left\| S \circ D \circ G\left(C^i\right) - C^i \right\|^4$$

$$\min_{\alpha} \left(S \circ D \circ G\left(C^i\right) - C^i\right)^t \left(S \circ D \circ G\left(C^i(\alpha)\right) - C^i(\alpha)\right) - \beta\alpha(1-\alpha)\left\| S \circ D \circ G\left(C^i\right) - C^i \right\|^2$$

$$\min_{\alpha} \ln\left(\left(\left(S \circ D \circ G\left(C^i\right) - C^i\right)^t \left(S \circ D \circ G\left(C^i(\alpha)\right) - C^i(\alpha)\right) \right)^2 \right) - \beta \ln\left(\alpha(1-\alpha)\left\| S \circ D \circ G\left(C^i\right) - C^i \right\|^4 \right)$$

$$\min_{\alpha} \ln\left(\left(\left(S \circ D \circ G\left(C^i\right) - C^i\right)^t \left(S \circ D \circ G\left(C^i(\alpha)\right) - C^i(\alpha)\right) \right)^2 \right)$$
$$- \beta \ln\left(\alpha\left\| S \circ D \circ G\left(C^i\right) - C^i \right\|^2 \right) \ln\left((1-\alpha)\left\| S \circ D \circ G\left(C^i\right) - C^i \right\|^2 \right)$$

Dataset Modeling

In order to activate this type of averaging, set `Stepsize computation option` in `Run Control File` (for example, `SDG-test.run`) to `6` (or `7` or `8` or `9` or `10` or `11` or `12` or `15` or `16` or `17`). Also, specify the `Stepsize computation parameter` in `Run Control File`. The following is the mapping between the value of `Stepsize computation option` and the respective potential:

6	$\min_{\alpha} \left\| S \circ D \circ G\left(C^i(\alpha)\right) - C^i(\alpha) \right\|^2 - \beta\alpha(1-\alpha)\left\| S \circ D \circ G\left(C^i\right) - C^i \right\|^2$
7	$\min_{\alpha} \left\| S \circ D \circ G\left(C^i(\alpha)\right) - C^i(\alpha) \right\|^2 - \beta\alpha^2\left\| S \circ D \circ G\left(C^i\right) - C^i \right\|^2$
15	$\min_{\alpha} \left\| S \circ D \circ G\left(C^i(\alpha)\right) - C^i(\alpha) \right\| - \beta\alpha(1-\alpha)\left\| S \circ D \circ G\left(C^i\right) - C^i \right\|$
16	$\min_{\alpha} \ln\left(\left\| S \circ D \circ G\left(C^i(\alpha)\right) - C^i(\alpha) \right\|^2 \right) - \beta \ln\left(\alpha(1-\alpha)\left\| S \circ D \circ G\left(C^i\right) - C^i \right\|^2 \right)$
17	$\min_{\alpha} \ln\left(\left\| S \circ D \circ G\left(C^i(\alpha)\right) - C^i(\alpha) \right\|^2 \right) - \beta \ln\left(\alpha\left\| S \circ D \circ G\left(C^i\right) - C^i \right\| \right) \ln\left((1-\alpha)\left\| S \circ D \circ G\left(C^i\right) - C^i \right\| \right)$
8	$\min_{\alpha} \left(\left(S \circ D \circ G\left(C^i\right) - C^i\right)^t \left(S \circ D \circ G\left(C^i(\alpha)\right) - C^i(\alpha)\right) \right)^2 - \beta\alpha(1-\alpha)\left\| S \circ D \circ G\left(C^i\right) - C^i \right\|^4$
9	$\min_{\alpha} \left(\left(S \circ D \circ G\left(C^i\right) - C^i\right)^t \left(S \circ D \circ G\left(C^i(\alpha)\right) - C^i(\alpha)\right) \right)^2 - \beta\alpha^2\left\| S \circ D \circ G\left(C^i\right) - C^i \right\|^4$

10	$\min_{\alpha}\left(S\circ D\circ G\left(C^i\right)-C^i\right)^t\left(S\circ D\circ G\left(C^i(\alpha)\right)-C^i(\alpha)\right)-\beta\alpha(1-\alpha)\left\Vert S\circ D\circ G\left(C^i\right)-C^i\right\Vert^2$
11	$\min_{\alpha}\ln\left(\left(\left(S\circ D\circ G\left(C^i\right)-C^i\right)^t\left(S\circ D\circ G\left(C^i(\alpha)\right)-C^i(\alpha)\right)\right)^2\right)-\beta\ln\left(\alpha(1-\alpha)\left\Vert S\circ D\circ G\left(C^i\right)-C^i\right\Vert^4\right)$
12	$\min_{\alpha}\ln\left(\left(\left(S\circ D\circ G\left(C^i\right)-C^i\right)^t\left(S\circ D\circ G\left(C^i(\alpha)\right)-C^i(\alpha)\right)\right)^2\right)$ $-\beta\ln\left(\alpha\left\Vert S\circ D\circ G\left(C^i\right)-C^i\right\Vert^2\right)\ln\left((1-\alpha)\left\Vert S\circ D\circ G\left(C^i\right)-C^i\right\Vert^2\right)$

Convergence

Algorithm

Guidance was generated by approximating a fixed point of the composite map. Progress of the fixed point algorithm towards convergence was measured using a link time inconsistency norm, defined as

$$IC_C=\left[\sum_l\sum_t\left\{C_l(t)-S\circ D\circ G(C(.))\right\}^2\right]^{1/2}$$

where, clearly the inconsistency norm attains the value 0 at a deterministic fixed point of the S◦D◦G : C → C map. Convergence to a constant but non-zero value could be (but is not necessarily) an indication that the stochastic process of link condition trajectories generated by successive iterations of the solution algorithm has become stationary (Bottom, 2000).

The presence of uninformed drivers who do not respond to the updated path splits prevents the norm from reaching 0; however, when the convergence curve becomes approximately horizontal, it can be concluded that convergence has been approximately reached (Bottom, Kachani, Perakis, 2006).

Code Implementation

- Store C^i to `ltt0`, in case of SDG mapping.
- Simulate the mapping and store $S\circ D\circ G\left(C^i\right)$ to `ltt1`, in case of SDG mapping.
- Call `link_time_table_norm`, in case of SDG mapping (or `path_split_table_norm`, in case of DGS mapping, or `msg_norm`, in case of GSD mapping) in order to get `norm`.

```
norm = link_time_table_norm (ltt0, ltt1);                    // || Ck - SDG(Ck) ||
ps_norm = path_split_table_norm (psp0, psp1);    // || Pk - DGS(Pk)||
m_norm = msg_norm (msg0, msg1);     // || Mk - GSD(Mk) ||
```

- ```
  float link_time_table_norm (link_time_table * ltt1, link_time_table * ltt2)
  {
    if (! (ltt1->granularity == ltt2->granularity && ltt1->nslices == ltt2->nslices)) { //mw2221
  sanity check
      cerr << "Attempt to compute norms of incommensurate link time tables !" << endl;
      exit (1);
    }
    double norm = 0;
    int nl = nw->get_nlinks (), ns = ltt1->nslices, gran = ltt1->granularity;
    //mw2221 the following lines calculate norm value
    for (int il = 0; il < nl; il++) //mw2221 for each links...
      for (int ip = 0; ip < ns; ip++) { //mw2221 for each period...
        double x = (abs(ltt1->lttab[il][ip]) - abs(ltt2->lttab[il][ip]));
        x /= time_scale; //mw2221 this makes sure that x is in tenths of second
        norm += x * x * gran; //mw2221 ???why not x*gran*x*gran = x*x*gran*gran???
      }
    return sqrt (norm);
  }
  #if 1
    for (int in = 0; in < nn; in++)
      if (pst1->enroutetab[in]) {
        for (int id = 0; id < nz; id++)
          for (int fp = 0, np = ph->get_npaths (in, id); fp < np; fp++)
            for (int tp = 0; tp < np; tp++)
              for (int it = 0; it < nperiods; it++) {
                double x =
                  pst1->enroutetab[in][id][fp][tp][it] - pst2->enroutetab[in][id][fp][tp][it];
                norm += x*x;
              }
      }
  #endif

    return sqrt (norm);
  }
  ```
- ```
  float path_split_table_norm
  ```

```
  (const path_split_table * pst1, const path_split_table * pst2)
{
  double norm = 0.;
  int nz = nw->get_nzones (), nn = nw->get_nnodes ();

  for (int io = 0; io < nz; io++)
    if (pst1->pretriptab[io]) {
      for (int ic = 0; ic < 2; ic++)
        for (int id = 0; id < nz; id++)
          for (int ip = 0, np = ph->get_npaths (io, id); ip < np; ip++)
            for (int it = 0; it < ninflowperiods; it++) {
              double x =
                pst1->pretriptab[io][ic][id][ip][it] - pst2->pretriptab[io][ic][id][ip][it];
              norm += x*x;
            }
    }
```

-
```
float msg_norm (const info_system * msg1, const info_system * msg2) {
  double norm = 0.;
  int nz = nw->get_nzones (), nl = nw->get_nlinks ();

  int nper = ninflowperiods;
  for (int io = 0; io < nz; io++)      // pre-trip msgs at origins
    if (msg1->pretripinfo[io] != NULL) {
      ptinfo * pt1 = msg1->pretripinfo [io][(int) FULL_GUIDANCE];
      ptinfo * pt2 = msg2->pretripinfo [io][(int) FULL_GUIDANCE];
      for (int iper = 0; iper < nper; iper++)
        for (int id = 0; id < nz; id++)
          for (int np = ph->get_npaths (io, id), ip = 0; ip < np; ip++) {
            double xxx = pt1->get_one_path_info (iper, id, ip) -
                         pt2->get_one_path_info (iper, id, ip);
            norm += xxx * xxx;
          }
    }

  nper = nperiods;
  for (int il = 0; il < nl; il++)       // en route msgs at beacons
    if (msg1->vms[il] != NULL) {
      ptinfo * pt1 = msg1->vms[il];
      ptinfo * pt2 = msg2->vms[il];
      int io = nw->get_bnode (il);
      for (int iper = 0; iper < nper; iper++)
        for (int id = 0; id < nz; id++)
          for (int np = ph->get_npaths (io, id), ip = 0; ip < np; ip++) {
            double xxx = pt1->get_one_path_info (iper, id, ip) -
                         pt2->get_one_path_info (iper, id, ip);
            norm += xxx * xxx;
          }
    }
  return sqrt (norm);
}
```

Exit Capacity and Storage Capacity

Algorithm

When a vehicle enters a link, its earliest possible exit time is calculated from the link's length and fixed speed; no account is taken in this calculation of other vehicles on the link. The vehicle is then placed at the tail of the link's queue and the link's available storage capacity (calculated from its length and number of lanes) is reduced accordingly. As each successive vehicle at the head of the queue is processed and moves on the downstream link, each following vehicle advances in position until it too arrives at the queue head.

For a vehicle at the head of a queue to advance to the next link, it must (i) be able to leave its current link and (ii) be accepted on the next link. A vehicle is only able to leave the current link if its earliest possible exit time is less than or equal to the current simulation time, and if the link that it is on has unused exit capacity remaining in the current time step. If these conditions are met, and if the link is an enroute decision point, the vehicle individually reselects a path in accordance with the input path splits corresponding to that location and the current time step; otherwise it retains its current path. In either case, the next link on the vehicle's path is determined.

An exiting vehicle advances if the next link on its path has storage capacity available, or if it has arrived at its destination.(Bottom, 2000).

Code Implementation

As defined in `vehicle_mover.cc`, at every period the algorithm visit each link and verify whether *exit capacity* is still available. If so, the algorithm further verifies whether there is a vehicle that is ready to be moved (vehicle's `earliest_link_exit_period > curr_period`). If so, incorporating the path splits, the algorithm tries to move the vehicle to the next link. Before doing so, the algorithm first verifies whether the downstream link has sufficient *storage capacity* for an additional vehicle. If so, the algorithm moves the vehicle.

Defined in `nw.dat`, the following is the setup in the original dataset:

"from" Node	"to" Node	Length (km)	Speed	Number of	Exit

			(km/hr)	Lanes	Capacity (vehicle/hr)
		L	V		C
1	100	0	100	1	10800
100	101	1	100	1	3600
101	102	1	100	1	3600
102	200	1	100	1	3600
100	201	1.5	100	1	3600
201	200	1.5	100	1	3600
100	301	1	100	1	3600
301	302	1	100	1	3600
302	200	1	100	1	3600
101	201	1	100	1	3600
201	302	1	100	1	3600
301	201	1	100	1	3600
201	102	1	100	1	3600
200	2	0	100	1	39600

EXIT CAPACITY

In `lad.cc`, the exit capacity of a particular link is calculated as follows:

"Exit Capacity" `= C *` "Number of Lanes" `*` "Flow Scale"

"Flow Scale" is defined in the Run Control File. It is the value of parameter `Trip scale factor`. The original dataset defines `Trip scale factor` as 1 for simplicity.

The unit of "Exit Capacity" is vehicle/hour. However, the time unit in the algorithm is "period" instead of "hour". Therefore, the exit capacity (c) is calculated as follows:

`c =` "Exit Capacity" `/ 3600 * sec_per_period`

`sec_per_period` is defined in the Run Control File. It is the value of parameter `Seconds per period`. The original dataset defines `Seconds per period` as 1 for simplicity.

Clearly the unit of `c` is vehicle/period. If `sec_per_period` is one then the unit is practically vehicle/second.

STORAGE CAPACITY

In `lad.cc`, the storage capacity of a particular link is calculated as follows:

"Storage Capacity" `= 1000 *` "Number of Lanes" `* L *` "Flow Scale" `/ AVG_VEH_LEN`

"Flow Scale" is defined in the Run Control File. It is the value of parameter `Trip scale factor`. The original dataset defines `Trip scale factor` as 1 for simplicity.

`AVG_VEH_LEN` (in meters) is defined in `simulation.h`. It is the average length of vehicles. The original code defines `AVG_VEH_LEN` as 7.5 (meters).

Dataset Modeling

Using the format described above, specify the network details in `nw.dat`.
"Flow Scale" is defined in the Run Control File. Specify the value as needed.
`sec_per_period` is defined in the Run Control File. Specify the value as needed.

`AVG_VEH_LEN` (in meters) is defined in `simulation.h`. Modifying this value implies modifying the whole code. Make sure that you re-compile the application after modifying `simulation.h`.

Rounding

Algorithm

Rounding is one of the sources of stochasticity in this application.

In general terms, stochasticity in the model outputs results from (i) randomizing the order in which trip departures and links are processed, so as to avoid systematic biases in model output, and (ii) rounding to integers any non-integer numbers of vehicles, link traversal times, flow rates or link capacities per time step.

Although the general idea is always to convert non-integer values to integers, different requirements and procedures apply to each case:

- Scalar Rounding
  - o to covert computer average link traversal times to the nearest time unit
- Vector Rounding
  - o bucket rounding method of vector rounding
  - o to determine the number of vehicles departing from an origin in each time step
  - o to determine link exit capacities per time step
- Scalar and vector rounding of link capacities per time step can lead to unsatisfactory results. Converting an exit capacity of 1799 vehicles/hour into per second integer equivalent in this way, for example, would block the link. To avoid this problem, the simulator implements *random rounding* of link exit capacities when scalar rounding is chosen.

Code Implementation

Rounding in `link_time_table`

in `link_time_table::read_link_time_file`

```
...
lttab[il][it] = (PERIOD) (time_scale * val + 0.5)
...
```

in `average_link_time_tables`

```
...
ltt1->lttab[il][ip] = (PERIOD) ((1.0-alpha) * abs(ltt1->lttab[il][ip]) + alpha * abs(ltt2-
>lttab[il][ip]) + 0.5);
...
```

in `link_time_tracker::compute_link_times`

`...` `ltt->lttab[il][ip] = (int) ((float) cum_time[il][ip] / (float) cum_entries[il][ip] + 0.5);`	populate corresponding cell of link time table with cum_time/cum_entries

`...`
`ltt->lttab[il][ip] =  int (ltt->lttab[il][sm-1] + step * (ip-sm+1) + 0.5);`
`...`
	interpolate link time

Rounding in `path_time_table`

in `link_time_table::compute_path_time`

```
...
t = (PERIOD) (t / time_scale + 0.5)
...
```

Rounding in `lad` (link attribute data)

in `lad::read_attribute_data`

`...` `if (bucket_round_link_exit_capacities) {`	
`int span = int (3600. / sec_per_period + 0.5);`	calculate the number of periods in one hour
`int r = int (span * U01 ());`	get random starting point during hour period U01 () returns uniform random variable between 0 and 1
`float c = lav[il].exit_cap / 3600. * sec_per_period;`	calculate exit capacity per period if sec_per_period is 1 then c is practically the exit capacity per second
`float dlt = 0.;`	The following lines calculate the delta that will make sure that exit_cap is retained despite of roundings

```
      for (int j = r; j < span; j++) {


        int z = int (c + dlt + 0.5);
        dlt += c - z;
      }
      lav[il].delta = dlt;
    }
    Else lav[il].delta = 0.0;
    ...
```

loop starts from a random starting point during hour period to the total number of periods

in `move_vehicles`

```
    ...
    int_per_ex_cap = int (per_ex_cap + ld->lav[il].delta + 0.5);
    ...
    if ((frac_per_ex_cap > trip_eps) && (U01() < frac_per_ex_cap)) int_per_ex_cap++;
    ...
```

Rounding in odp (origin-destination-path)

in `odp::process_trip_specs`

```
    ...
    if ((frac_trips > trip_eps) && (U01() < frac_trips))
    trips++



    ...
```

create the required number of vehicles if fractional number of trips, avoid systematic bias from deterministic rounding

in `odp_flows::initialize_bucket_rounding`

```
    ...
  int span;
  If (next_time_period == MAX_PERIOD) span = int (3600. /
sec_per_period + 0.5);
  else span = next_time_period - curr_period;
  for (int it = 0; it < (signed) fsv.size (); it++) { //
for each current fsv
    int r = int (span * U01()); // pick random start

    if (r == span) r = span - 1;

    float t = fsv[it].ntrips / 3600. * sec_per_period; //
# trips in period

    float fdelta = fsv[it].delta; // initial discrepancy
should be 0

    for (int j = r; j < span; j++) {


      int z = int (t + fdelta + 0.5);
      fdelta = t + fdelta - z;
    }
    fsv[it].delta = fdelta;
    ...
```

calculate the number of periods in one hour

get random starting point during hour period

U01 () returns uniform random variable between 0 and 1

calculate number of trips per period; if sec_per_period is 1 then t is practically the number of trips per second

the following lines calculate the delta that will make sure that number of trips is retained despite of roundings

loop starts from a random starting point during hour period to the total number of periods

in `odp_flows::create_trips`

```
    ...
    nt = int (t + 0.5);
    ...
```

in `od_flows::create_trips`

```
    ...
    float odz = int (t + 0.5);
    ...
    float odpz = int (f[r] + delt + 0.5);
    ...
    float odpz = int (f[ip] + pdelta[ip] + 0.5);
    ...
    int odt = int (t + 0.5);
    ...
```

Rounding in `ptinfo`

in `average_msgs`

```
...
pt1->set_one_path_info (iper, id, ip, PERIOD (calpha * pt1->get_one_path_info (iper, id, ip)
+ alpha * pt2->get_one_path_info (iper, id, ip) + 0.5));
...
```

```
pt1->set_one_path_info (iper, id, ip, PERIOD (calpha * pt1->get_one_path_info (iper, id, ip)
+ alpha * pt2->get_one_path_info (iper, id, ip) + 0.5));
...
```

Basic Components of the Application

Input Data

Run Control File

Example:

```
[Trip scale factor]              1.0
[Time scale factor]             10.0
[Network file name]           nw.dat
[Path file name]              nw.pth
[Number of periods]             2400
[Seconds per period]               1
[Random number seed]            4163
[DBM aggregation]                  D
[Link capacity rounding]           B
[Processing option]                1
[Number of replications]           3
[Number of iterations]           100
[Stepsize computation option]      1
```

Format of the run control file:

	Description
Row 1	"flow scale" multiplier
Row 2	"time scale" multiplier
Row 3	network file name
Row 4	path file name
Row 5	max number of periods (will self-stop earlier if possible)
Row 6	seconds per period
Row 7	random number seed
Row 8	DBM application; possible values are `D` for disaggregate or `A` for aggregate
Row 9	link capacity rounding; possible values are `B` for bucket rounding or `R` for random rounding
Row 10	processing option
Row 11	number of replications
Row 12	number of iterations
Row 13	step size computation method option; possible values are as follows: `0` for constant step size averaging method; this method requires additional parameter, which is constant value that is specified in Row 14. `1` for method of successive averaging (MSA) `5` for polyak iterate averaging method; this method requires additional parameter, which is a multiplier that is specified in Row 14. `6` for line search method that optimizes potential as follows: $\min\limits_{\alpha}\left\|mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right\|^2-\beta\alpha(1-\alpha)\left\|mapping\left(C^i\right)-C^i\right\|^2$ This method requires additional parameter, which is a multiplier (β) that is specified in Row 14. `7` for line search method that optimizes potential as follows: $\min\limits_{\alpha}\left\|mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right\|^2-\beta\alpha^2\left\|mapping\left(C^i\right)-C^i\right\|^2$ This method requires additional parameter, which is a multiplier (β) that is specified in Row 14. `8` for line search method that optimizes potential as follows: $\min\limits_{\alpha}\left(\left(mapping\left(C^i\right)-C^i\right)^t\left(mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right)\right)^2-\beta\alpha(1-\alpha)\left\|mapping\left(C^i\right)-C^i\right\|^4$ This method requires additional parameter, which is a multiplier (β) that is specified in Row 14. `9` for line search method that optimizes potential as follows: $\min\limits_{\alpha}\left(\left(mapping\left(C^i\right)-C^i\right)^t\left(mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right)\right)^2-\beta\alpha^2\left\|mapping\left(C^i\right)-C^i\right\|^4$ This method requires additional parameter, which is a multiplier (β) that is specified in Row 14. `10` for line search method that optimizes potential as follows: $\min\limits_{\alpha}\left(mapping\left(C^i\right)-C^i\right)^t\left(mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right)-\beta\alpha(1-\alpha)\left\|mapping\left(C^i\right)-C^i\right\|^2$

This method requires additional parameter, which is a multiplier (β) that is specified in Row 14.

`11` for line search method that optimizes potential as follows:

$$\min_{\alpha} \ln\left(\left(\left(mapping\left(C^i\right)-C^i\right)^t\left(mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right)\right)^2\right)-\beta\ln\left(\alpha(1-\alpha)\left\|mapping\left(C^i\right)-C^i\right\|^4\right)$$

This method requires additional parameter, which is a multiplier (β) that is specified in Row 14.

`12` for line search method that optimizes potential as follows:

$$\min_{\alpha} \ln\left(\left(\left(mapping\left(C^i\right)-C^i\right)^t\left(mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right)\right)^2\right)$$
$$-\beta\ln\left(\alpha\left\|mapping\left(C^i\right)-C^i\right\|^2\right)\ln\left((1-\alpha)\left\|mapping\left(C^i\right)-C^i\right\|^2\right)$$

This method requires additional parameter, which is a multiplier (β) that is specified in Row 14.

`15` for line search method that optimizes potential as follows:

$$\min_{\alpha}\left\|mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right\|-\beta\alpha(1-\alpha)\left\|mapping\left(C^i\right)-C^i\right\|$$

This method requires additional parameter, which is a multiplier (β) that is specified in Row 14.

`16` for line search method that optimizes potential as follows:

$$\min_{\alpha} \ln\left(\left\|mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right\|^2\right)-\beta\ln\left(\alpha(1-\alpha)\left\|mapping\left(C^i\right)-C^i\right\|^2\right)$$

This method requires additional parameter, which is a multiplier (β) that is specified in Row 14.

`17` for line search method that optimizes potential as follows:

$$\min_{\alpha} \ln\left(\left\|mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right\|^2\right)$$
$$-\beta\ln\left(\alpha\left\|mapping\left(C^i\right)-C^i\right\|\right)\ln\left((1-\alpha)\left\|mapping\left(C^i\right)-C^i\right\|\right)$$

This method requires additional parameter, which is a multiplier (β) that is specified in Row 14.

Row 14	additional parameter

Network File

Example (`nw.dat`):

```
     2
     1   2
     1 100  0.0  100.0 1         10800.
   100 101  1.0  100.0 1          3600.
   101 102  1.0  100.0 1          3600.
   102 200  1.0  100.0 1          3600.
   100 201  1.5  100.0 1          3600.
   201 200  1.5  100.0 1          3600.
   100 301  1.0  100.0 1          3600.
   301 302  1.0  100.0 1          3600.
   302 200  1.0  100.0 1          3600.
   101 201  1.0  100.0 1          3600.
   201 302  1.0  100.0 1          3600.
   301 201  1.0  100.0 1          3600.
   201 102  1.0  100.0 1          3600.
   200   2  0.0  100.0 1         39600.
```

number of zones

origin node

destination node

2					
1	2				
1	100	0.0	100.0	1	10800
100	101	1.0	100.0	1	3600
...	...	...	...	...	...

"to" node

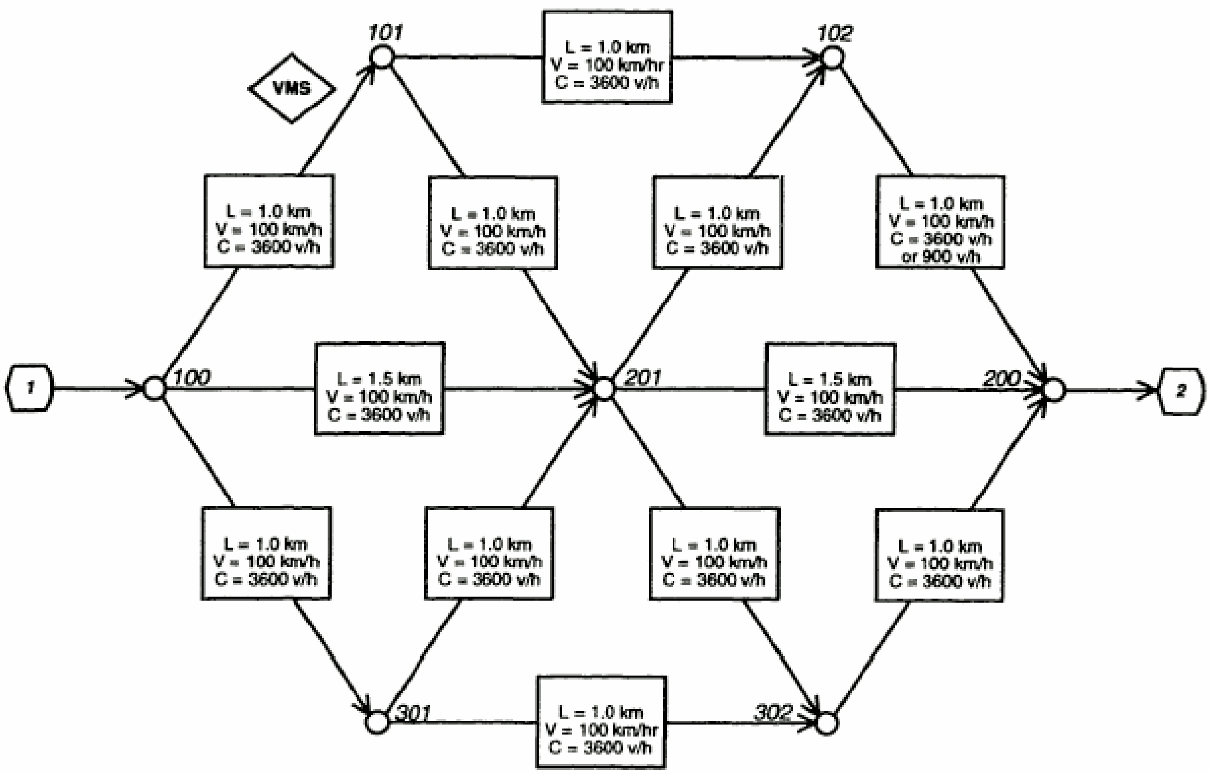
L in km

V in km/hr

number of lanes

C, exit capacity in vehicle/hr

"from" node



Path File

Example (`nw.pth`):

```
1  100  101 102 200       2  9999
1  100  101 201 102  200  2  9999
1  100  101 201 200       2  9999
1  100  101 201 302  200  2  9999
1  100  201 200           2  9999
1  100  201 102 200       2  9999
1  100  201 302 200       2  9999
1  100  301 302      200  2  9999
1  100  301 201 102  200  2  9999
1  100  301 201 200       2  9999
1  100  301 201 302  200  2  9999
```

flows.dat

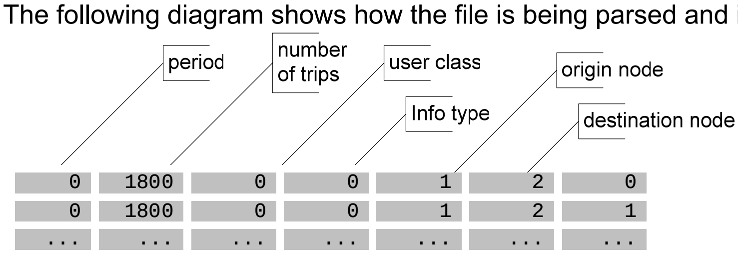
Flow rate file contains time period, flow rate (per hour), origin node, and destination node. The data are sorted by time period.

This input data file is hard-coded in `flowmov.cc`.

Example:

```
      0   1800   0 0   1 2    0
      0   1800   0 0   1 2    1
      0   1800   0 0   1 2    2
      0   1800   0 0   1 2    3
      0   1800   0 0   1 2    4
      0   1800   0 0   1 2    5
      0   1800   0 0   1 2    6
      0   1800   0 0   1 2    7
      0   1800   0 0   1 2    8
      0   1800   0 0   1 2    9
      0   1800   0 0   1 2   10
    240      0   0 0   0 0    0
```

The following diagram shows how the file is being parsed and interpreted by the application.



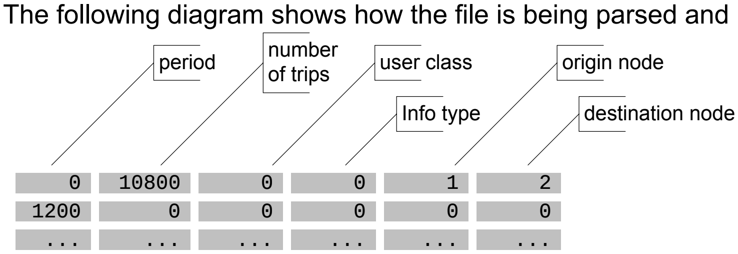
odflows.dat

This input data file is hard-coded in `DGS_main.cc`, `GSD_main.cc`, `SDG_main.cc`, and `eval_norm.cc`.

Example:

```
      0    10800    0 0 1 2
   1200        0    0 0 0 0
```

The following diagram shows how the file is being parsed and interpreted by the application.



Other Hard-Coded Constants

alpha

This constant is hard-coded in `path_split_model.cc`.

AVG_VEH_LEN

This constant is hard-coded in `simulation.h`.

g_npers

This constant is hard-coded in `DGS_gibbs.cc`.

g_nvper

This constant is hard-coded in `DGS_gibbs.cc`.

gran

This constant is hard-coded in `DGS_gibbs.cc`, `DGS_main.cc`, `GSD_main.cc`, `SDG_main.cc`, `SDG_ppp.cc`, `eval_norm.cc`.

integer_trips

This constant is hard-coded in `simulation.cc`.

MAX_FILE_NAME_LENGTH

This constant is hard-coded in `simulation.cc`.

nvmslinks

This constant is hard-coded in `ptinfo.cc`.

The original code hard-code the number of VMS links as well as the specific nodes of the link, node 100 and node 101. This setting works specifically with the data set that came with the original code.

start_iter

This constant is related to the implementation of polyak iterate averaging method (Bottom, 2000, page 136):

> When applied to find a fixed point of the $S \circ D \circ G$ map, the polyak algorithm can be written:
>
> $$C^{i+1} = C^{i} + \alpha^{i}\left(S \circ D \circ G\left(C^{i}\right) - C^{i}\right);\ i = 0\ldots$$
>
> $$\tilde{C}^{i+1} = \tilde{C}^{i} + \frac{1}{i+1} C^{i+1}$$
>
> where $\tilde{C}^{n}$ in the final iteration n is the fixed point estimate.
>
> In practice, iterate averaging (computation of the $\tilde{C}^{i}$ s) is only started after the MSA-like step shows signs of stabilizing. (This is called the *window of averaging*.)

Therefore, `start_iter` is an approximation of when MSA-like step shows signs of stabilizing.
This constant is hard-coded in `DGS_main.cc`, `GSD_main.cc`, `SDG_main.cc`, `SDG_ppp.cc`.

switching_penalty

This constant is hard-coded in `path_split_model.cc`.

trip_eps

This constant is hard-coded in `simulation.h`.

two_thirds

This constant is hard-coded in `DGS_main.cc`, `GSD_main.cc`, `SDG_main.cc`, `SDG_ppp.cc`.

**_output*

These constants are:

```
logfile_output
convfile_output
link_vol_output
in_link_time_output
path_vol_output
path_time_output
veh_event_output
path_switch_output
in_path_split_output
out_link_time_output
out_path_split_output
```

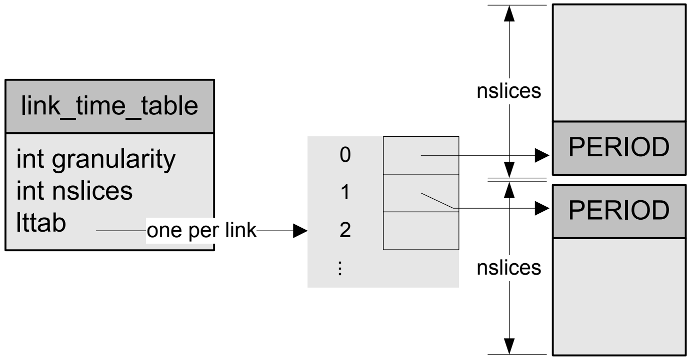
```
in_msg_output
out_msg_output
```

These constants are hard-coded in `simulation.cc`:

Data Structure

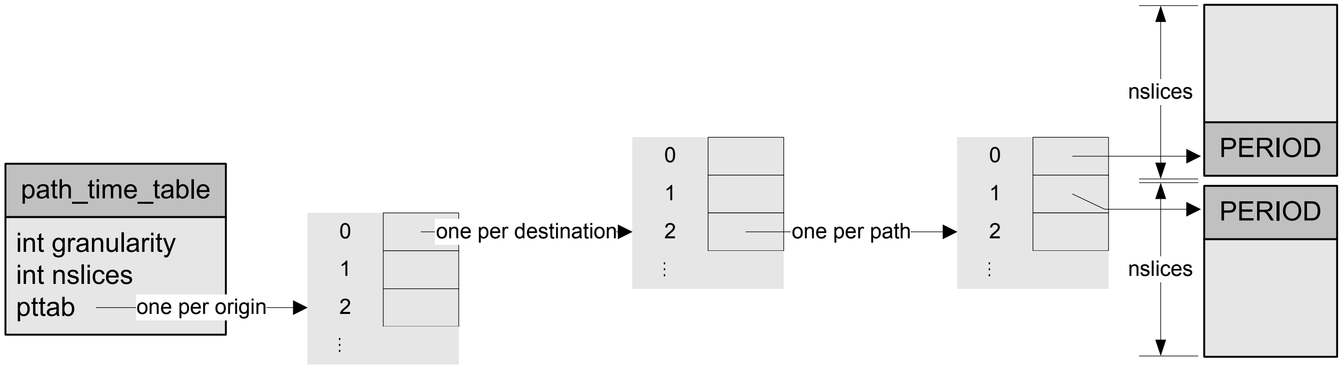
link_time_table

The following diagram shows the graphical representation of the data structure:



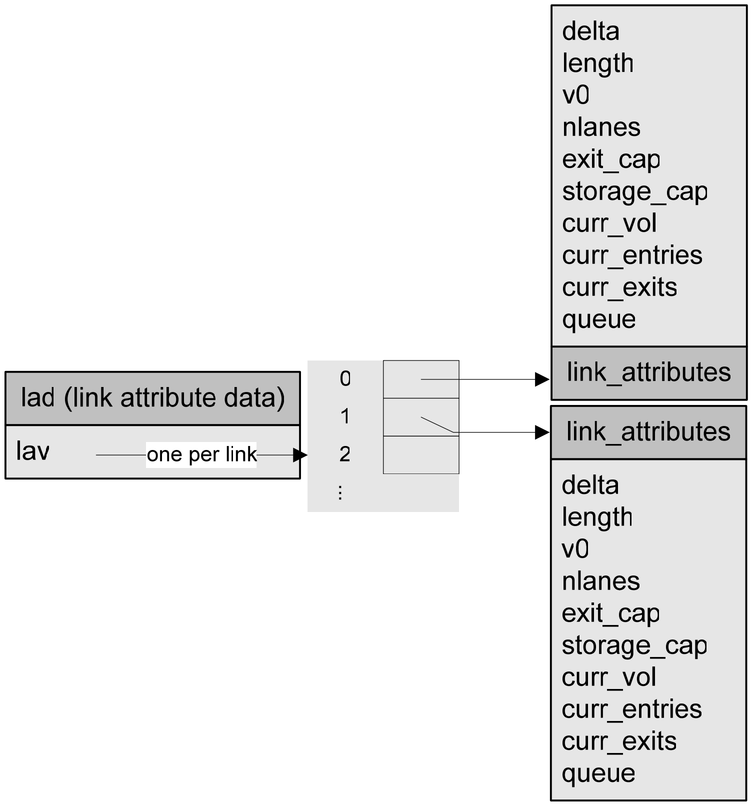
path_time_table

The following diagram shows the graphical representation of the data structure:



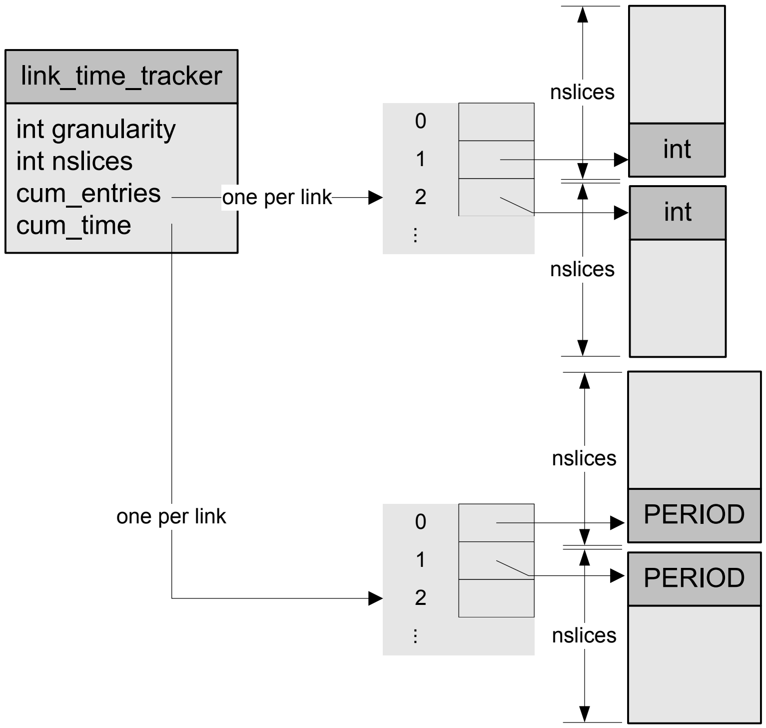
lad

The following diagram shows the graphical representation of the data structure:



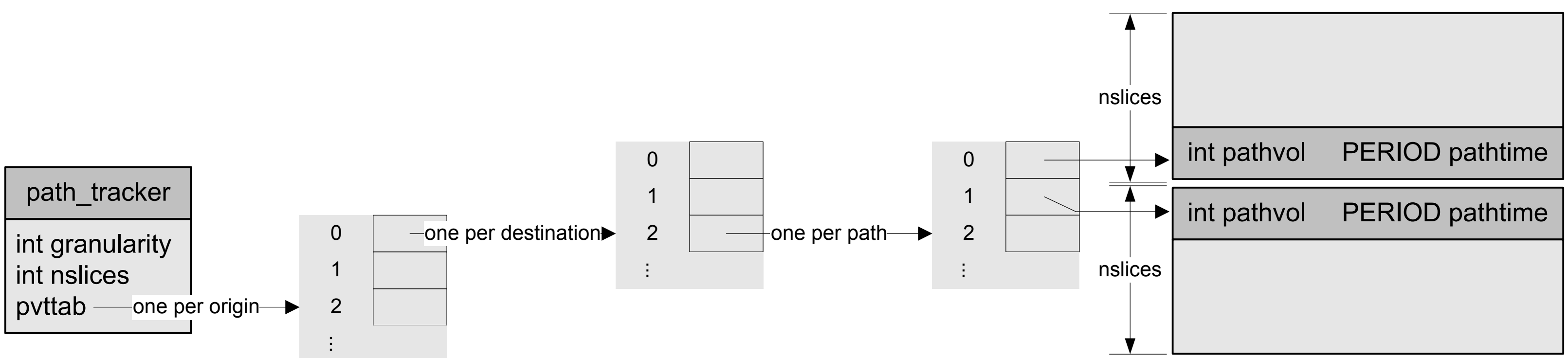
link_time_tracker

The following diagram shows the graphical representation of the data structure:



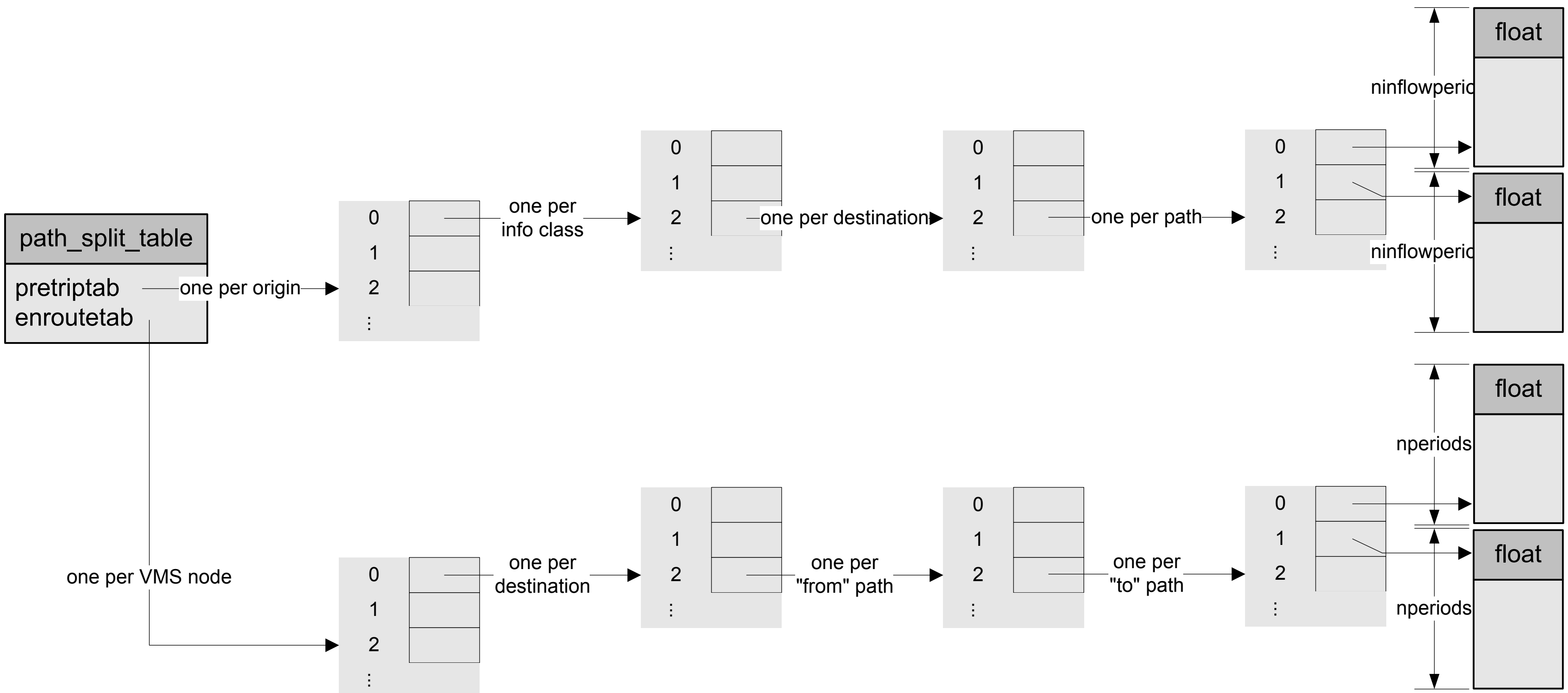
path_tracker

The following diagram shows the graphical representation of the data structure:



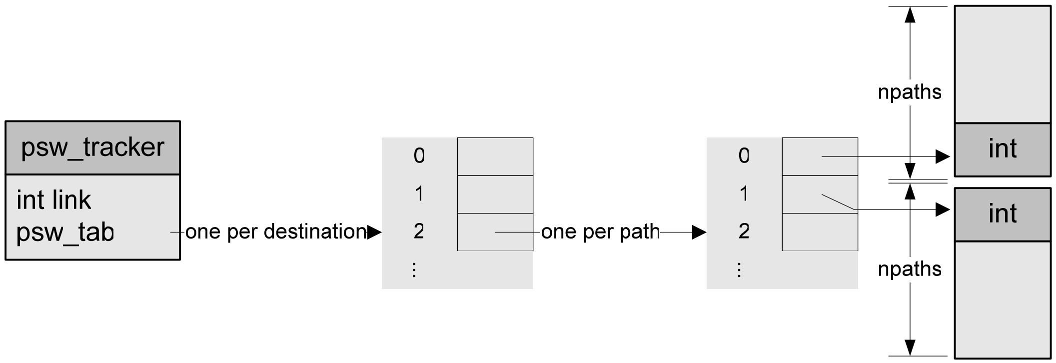
path_split_table

The following diagram shows the graphical representation of the data structure:



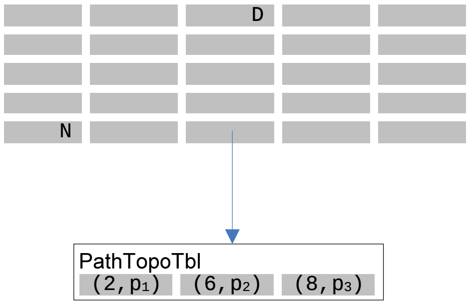
psw_tracker

The following diagram shows the graphical representation of the data structure:

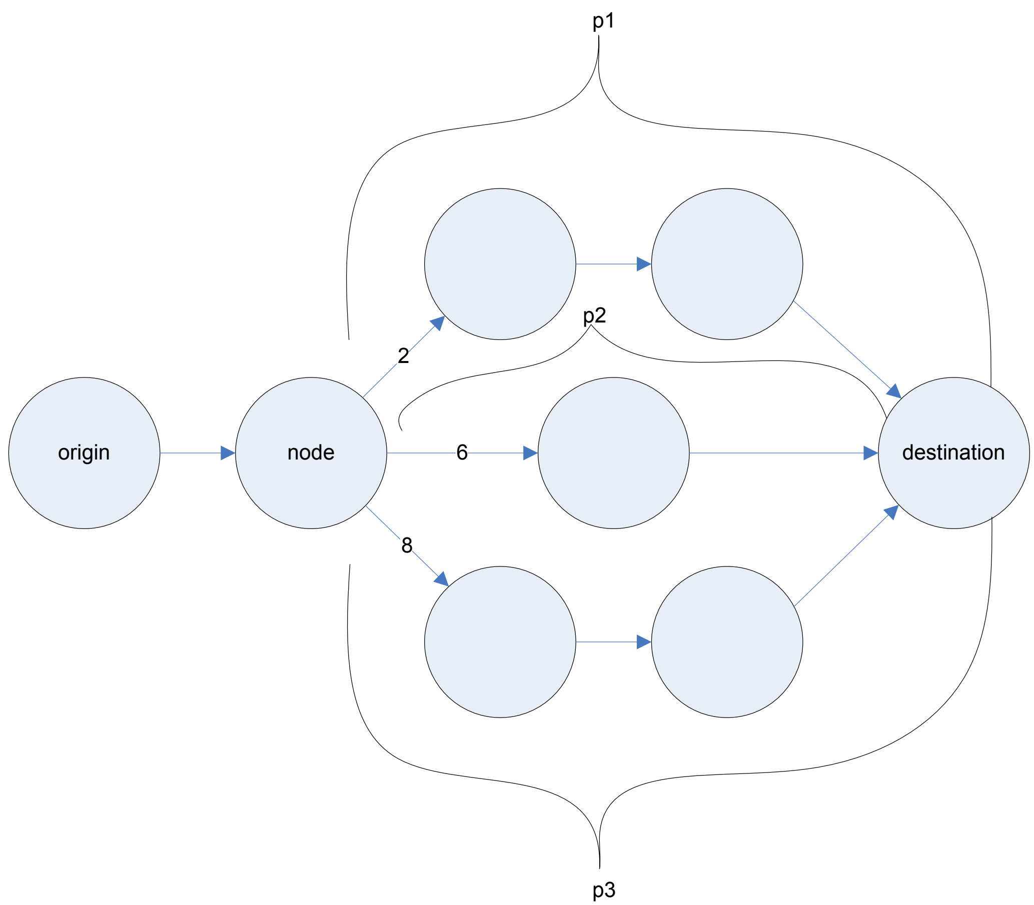


node-destination-path (ndp)

The following diagram shows the logical representation of the data structure:

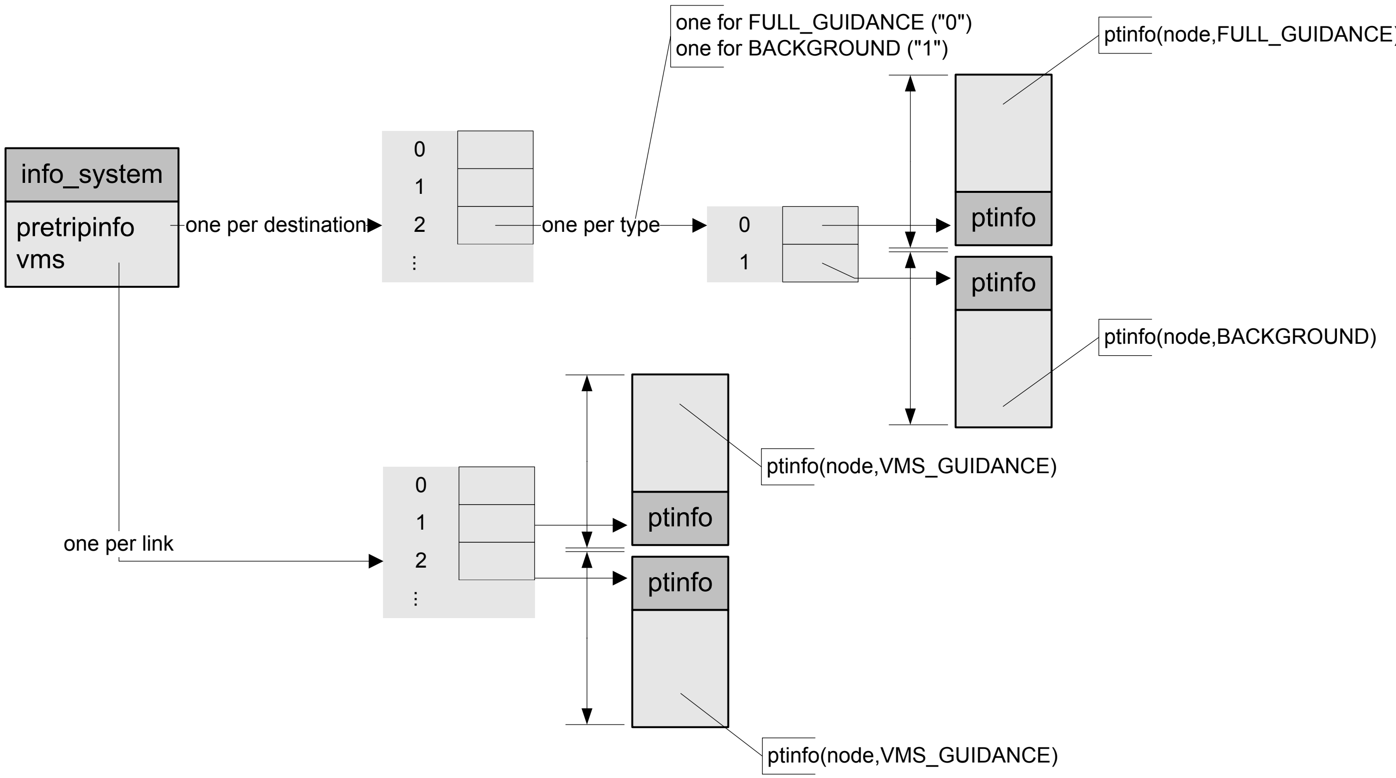


The above example of `PathTopoTbl` should be logically interpreted as follows:


p1
p2
2
origin
node
6
destination
8
p3

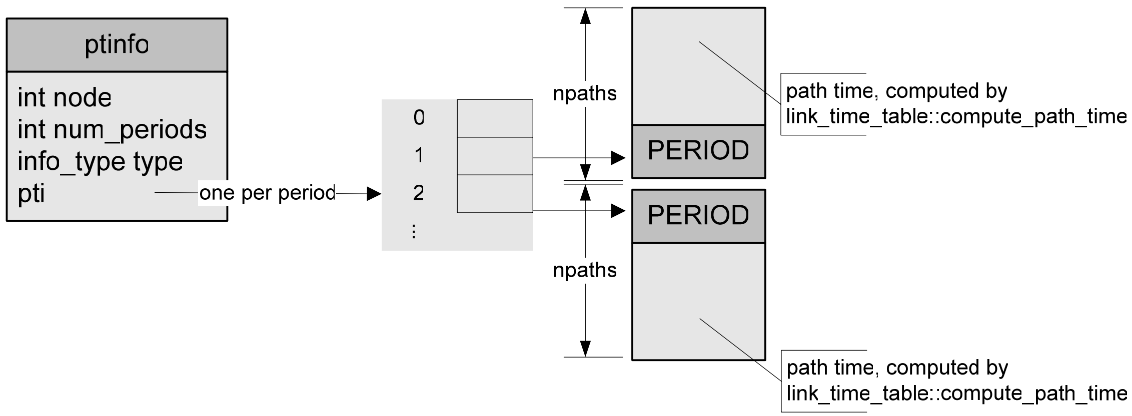
info_system

The following diagram shows the graphical representation of the data structure:



ptinfo

The following diagram shows the graphical representation of the data structure:



Output

log file

The file name follows the naming convention <test case name>`.log.out`. This file records the progress of the simulation.

convergence file

The file name follows the naming convention <test case name>`.cnv.out`.

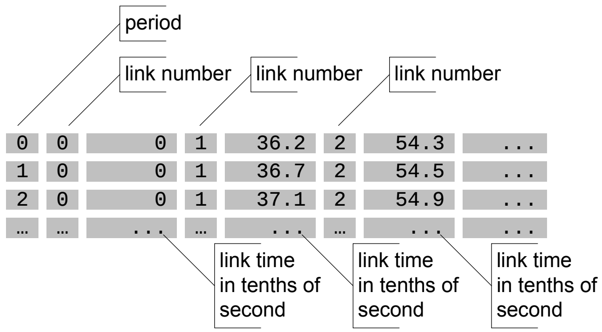
In the case of testing the SDG mapping, the columns of the file consist of iteration number in the first column, link time inconsistency norm in the second and third columns (the content of these two columns are identical), and the value of alpha in the fourth column. In the case of testing the DGS mapping, the columns of the file consist of iteration number in the first columns, path split inconsistency norm in the second column, link time inconsistency norm in the third column, and the value of alpha in the fourth column.

File Link Time Output (FLTO)

For example:

```
0 0 0 1 36.2 2 54.3 3 36.2 4 36 5 36 6 36 7 0 8 36 9 54 10 36 11 36 12 36 13 36
1 0 0 1 36.7 2 54.5 3 36.7 4 36 5 36 6 36 7 0 8 36 9 54 10 36 11 36 12 36 13 36
2 0 0 1 37.1 2 54.9 3 36.8 4 36 5 36 6 36 7 0 8 36 9 54 10 36 11 36 12 36 13 36
...
```

The following diagram shows how the file should be parsed and interpreted.

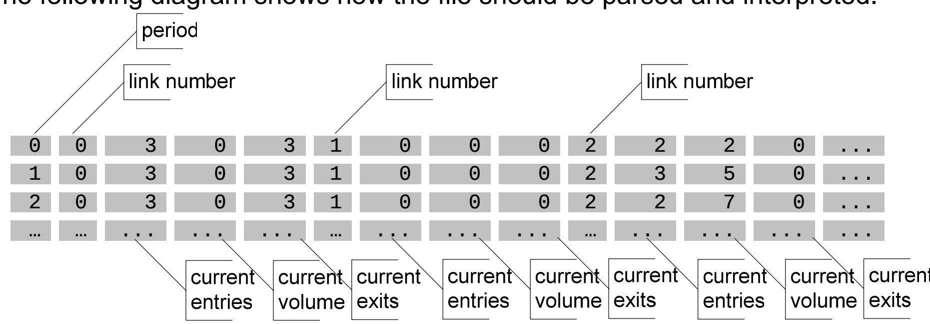


File Link Volumes Output (FLVO)

For example:

```
0 0 3 0 3 1 0 0 0 2 2 2 0 3 1 1 0 4 0 0 0 5 0 0 0 6 0 0 0 7 0 0 0 8 0 0 0 9 0 0 0 10 0 0 0 ...
1 0 3 0 3 1 0 0 0 2 3 5 0 3 0 1 0 4 0 0 0 5 0 0 0 6 0 0 0 7 0 0 0 8 0 0 0 9 0 0 0 10 0 0 0 ...
2 0 3 0 3 1 0 0 0 2 2 7 0 3 1 2 0 4 0 0 0 5 0 0 0 6 0 0 0 7 0 0 0 8 0 0 0 9 0 0 0 10 0 0 0 ...
...
```

The following diagram shows how the file should be parsed and interpreted.

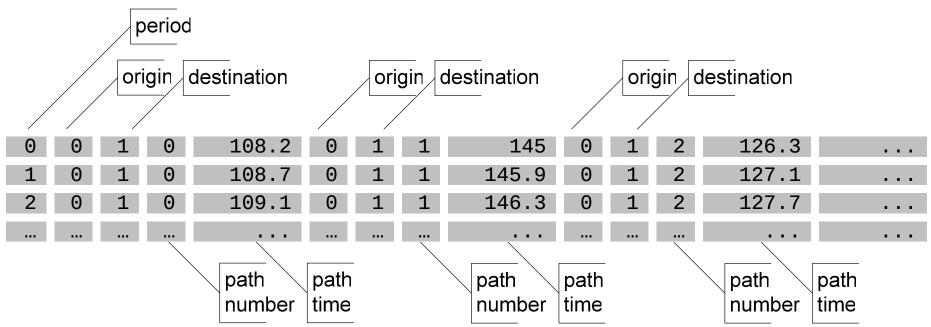


File Path Time Output (FPTO)

For example:

```
0 0 1 0 108.2 0 1 1 145 0 1 2 126.3 0 1 3 145.2 0 1 4 108.3 0 1 5 126.3 0 1 6 126.3 ...
1 0 1 0 108.7 0 1 1 145.9 0 1 2 127.1 0 1 3 145.7 0 1 4 108.5 0 1 5 126.6 0 1 6 126.7 ...
2 0 1 0 109.1 0 1 1 146.3 0 1 2 127.7 0 1 3 146.2 0 1 4 108.9 0 1 5 127.2 0 1 6 127.2...
...
```

The following diagram shows how the file should be parsed and interpreted.

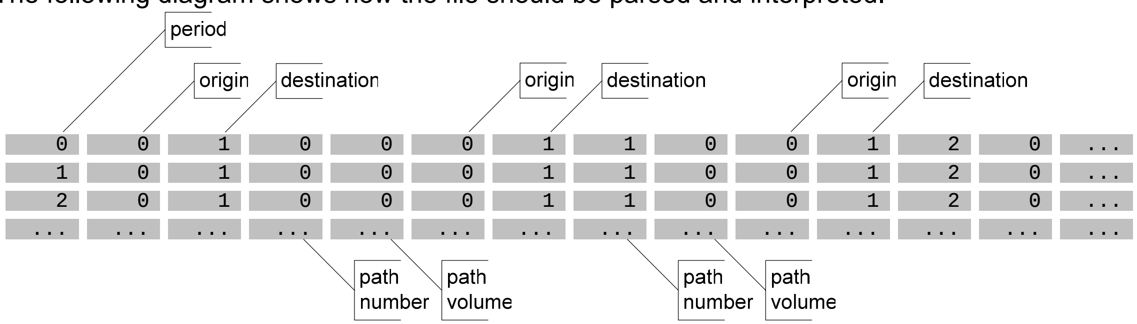


File Path Volumes Output (FPVO)

For example:

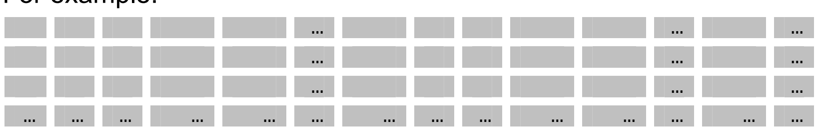
```
0 0 1 0 0 0 1 1 0 0 1 2 0 0 1 3 0 0 1 4 1 0 1 5 0 0 1 6 1 0 1 7 0 0 1 8 0 0 1 9 1 0 1 10 0
1 0 1 0 0 0 1 1 0 0 1 2 0 0 1 3 0 0 1 4 2 0 1 5 1 0 1 6 0 0 1 7 0 0 1 8 0 0 1 9 0 0 1 10 0
2 0 1 0 0 0 1 1 0 0 1 2 0 0 1 3 0 0 1 4 1 0 1 5 1 0 1 6 0 0 1 7 1 0 1 8 0 0 1 9 0 0 1 10 0
...
```

The following diagram shows how the file should be parsed and interpreted.

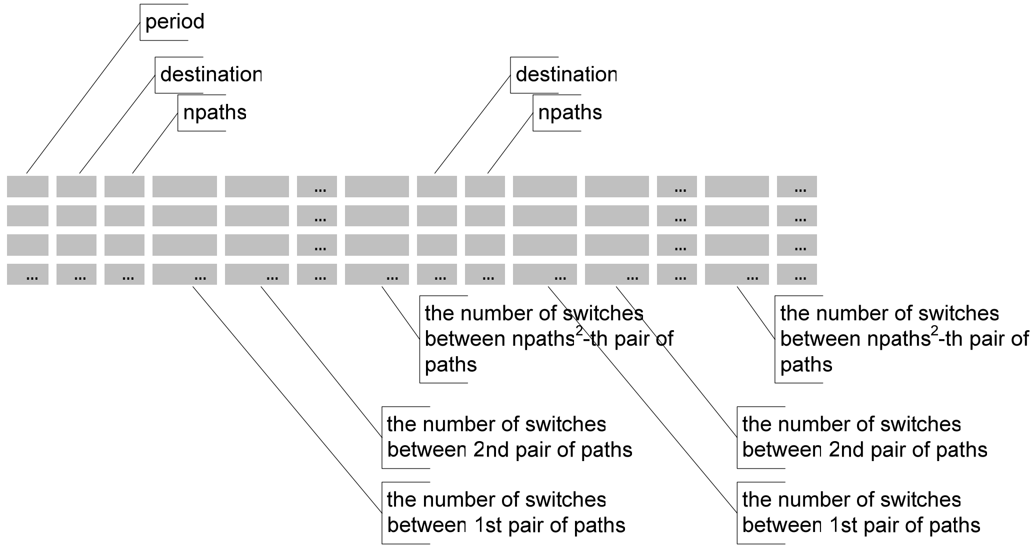


File Path Switches Output (FSWO)

For example:



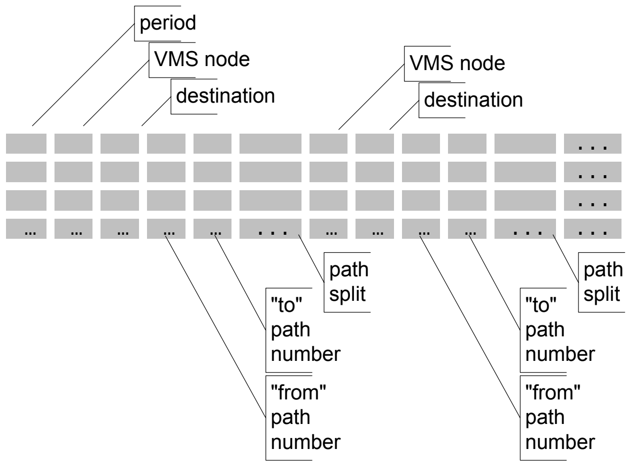
The following diagram shows how the file should be parsed and interpreted.



File Split Path Output (FSPO)

For example:

The following diagram shows how the file should be parsed and interpreted.



File Vehicle Events Output (FVEO)

For example:

	vehicle enters network: ID		user class		info class		origin		destination		initial path			
…	...	…	...	…	...	…	...	…	...	…	...	…		…
	vehicle exits network: ID													
…	...	…	...	…	...	…	...	…	...	…	...	…		…
	vehicle enters link: ID		link		(		-		)		earliest exit period			
…	...	…	...	…	...	…	...	…	...	…	...	…		…
	vehicle exits link: ID		link		(		-		)					
…	...	…	...	…	...	…	...	…	...	…	...	…		…
	vehicle ID		received guidance on link		(		-		)		old path :		- new path :	
…	...	…	...	…	...	…	...	…	...	…	...	…		…
	vehicle ID		blocked by spillback from link		(		-		)					
…	...	…	...	…	...	…	...	…	...	…	...	…		…

The following diagram shows how the file should be parsed and interpreted.

period

	vehicle enters network: ID		user class		info class		origin		destination		initial path			
…	...	…	...	…	...	…	...	…	...	…	...	…		…
	vehicle exits network: ID													
…	...	…	...	…	...	…	...	…	...	…	...	…		…
	vehicle enters link: ID		link		(		-		)		earliest exit period			
…	...	…	...	…	...	…	...	…	...	…	...	…		…
	vehicle exits link: ID		link		(		-		)					
…	...	…	...	…	...	…	...	…	...	…	...	…		…
	vehicle ID		received guidance on link		(		-		)		old path :		- new path :	
…	...	…	...	…	...	…	...	…	...	…	...	…		…
	vehicle ID		blocked by spillback from link		(		-		)					
…	...	…	...	…	...	…	...	…	...	…	...	…		…

"from" node

"to" node

Working with the Application

Prepare Development Environment

Install Java Platform

Download and install Java 2 Platform from URL http://java.sun.com/j2se/1.4.2/download.html. At the end of installation, make sure that environment variable `PATH` contains the path to `java` executable. This executable should be located under `bin` directory of Java 2 Platform installation environment.

Install CYGWIN

Download and install CYGWIN from URL http://www.cygwin.com/. At the end of installation, make sure that environment variable `PATH` contains the path to `gcc` executable. This executable should be located under `bin` directory of CYGWIN installation environment.

Install Eclipse

Download and install Eclipse from URL http://www.eclipse.org/downloads/. At the time when this report was written, the latest version of Eclipse was found at URL http://www.eclipse.org/downloads/download.php?file=/eclipse/downloads/drops/R-3.1.2-200601181600/eclipse-SDK-3.1.2-win32.zip.

Install C/C++ Development Tools (CDT)

Download and install C/C++ Development Tools (CDT) from URL http://download.eclipse.org/tools/cdt/releases/eclipse3.1/. At the time when this was written, the latest version of CDT was found at URL http://www.eclipse.org/downloads/download.php?file=/tools/cdt/releases/eclipse3.1/dist/3.0.2/org.eclipse.cdt-3.0.2-win32.x86.zip.
There is no automated installer for installing this component. Instead, one needs to uncompress the files and copy them to the proper directories of Eclipse installation environment.

Retrieve All Files from UNIX Environment

Retrieve the following files.

```
makefile
*.cc
*.h
```

There are 3 sets of code that were created respective to the particular test cases that were being observed. For example, if one wants to execute test case `2_0_10000_ITERATIONS`, then the set of code that needs to be retrieved will be `CODE_2_0_ORIGINAL_MIGRATED`.

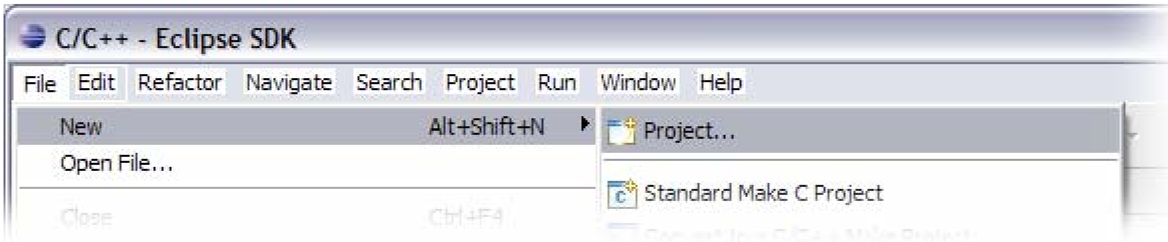
Set of Test Cases	Set of Code	Location in CD
2_0_10000_ITERATIONS	CODE_2_0_ORIGINAL_MIGRATED	<CD Root>/CODE_2_0_ORIGINAL_MIGRATED
2_0_RELAXED_EXIT_CAPACITY	CODE_2_0_ORIGINAL_MIGRATED	<CD Root>/CODE_2_0_ORIGINAL_MIGRATED
2_0_RELAXED_STORAGE_CAPACITY	CODE_2_0_ORIGINAL_MIGRATED	<CD Root>/CODE_2_0_ORIGINAL_MIGRATED
2_1_MSA_THEN_POLYAK	CODE_2_1_MSA_THEN_POLYAK	<CD Root>/CODE_2_1_MSA_THEN_POLYAK
2_2_POTENTIALS	CODE_2_2_POTENTIALS	<CD Root>/CODE_2_2_POTENTIALS

Locate a destination where the development project is going to reside. Copy all of the files to this location.

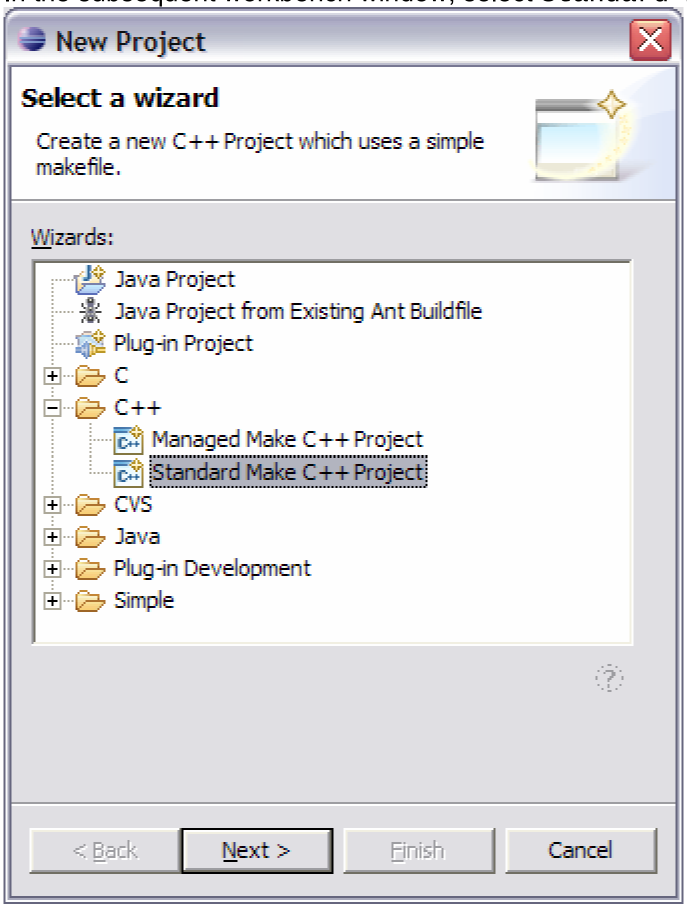
Create a Project

Bring up ECLIPSE and the following steps in order to create a C++ project.

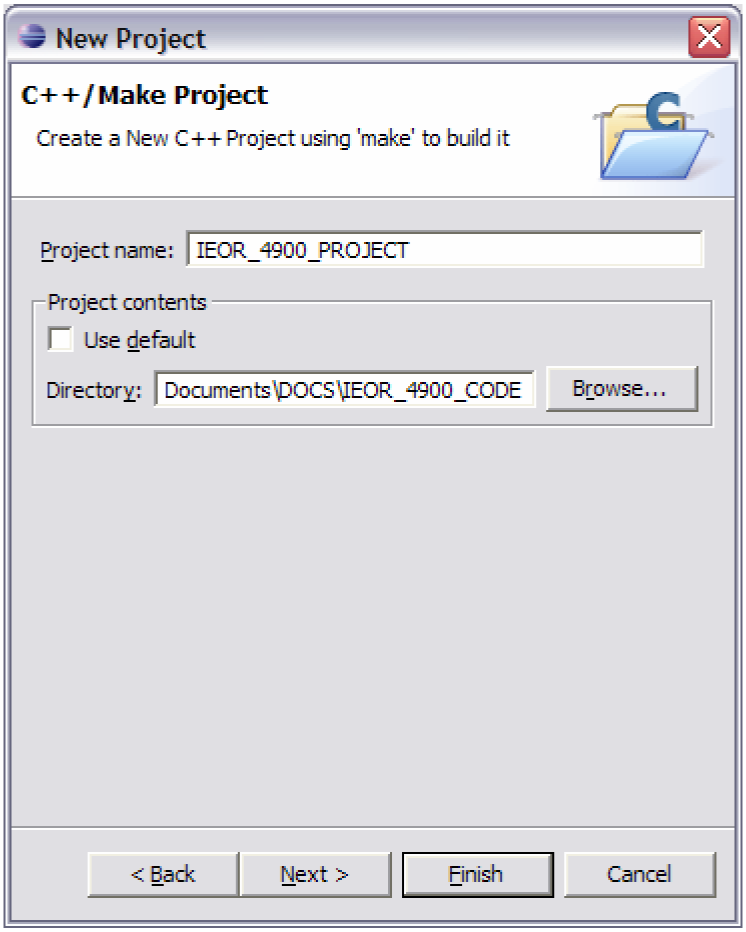
1. Select menu `File` > `New` > `Project...`.



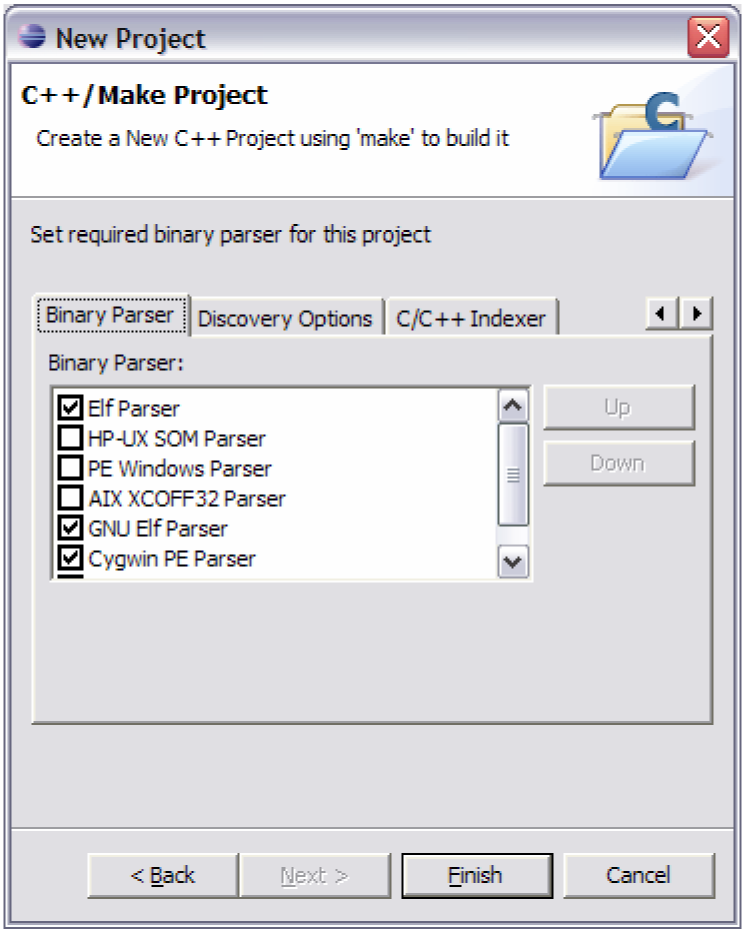
2. In the subsequent workbench window, select `Standard Make C++ Project`. Click `Next`.



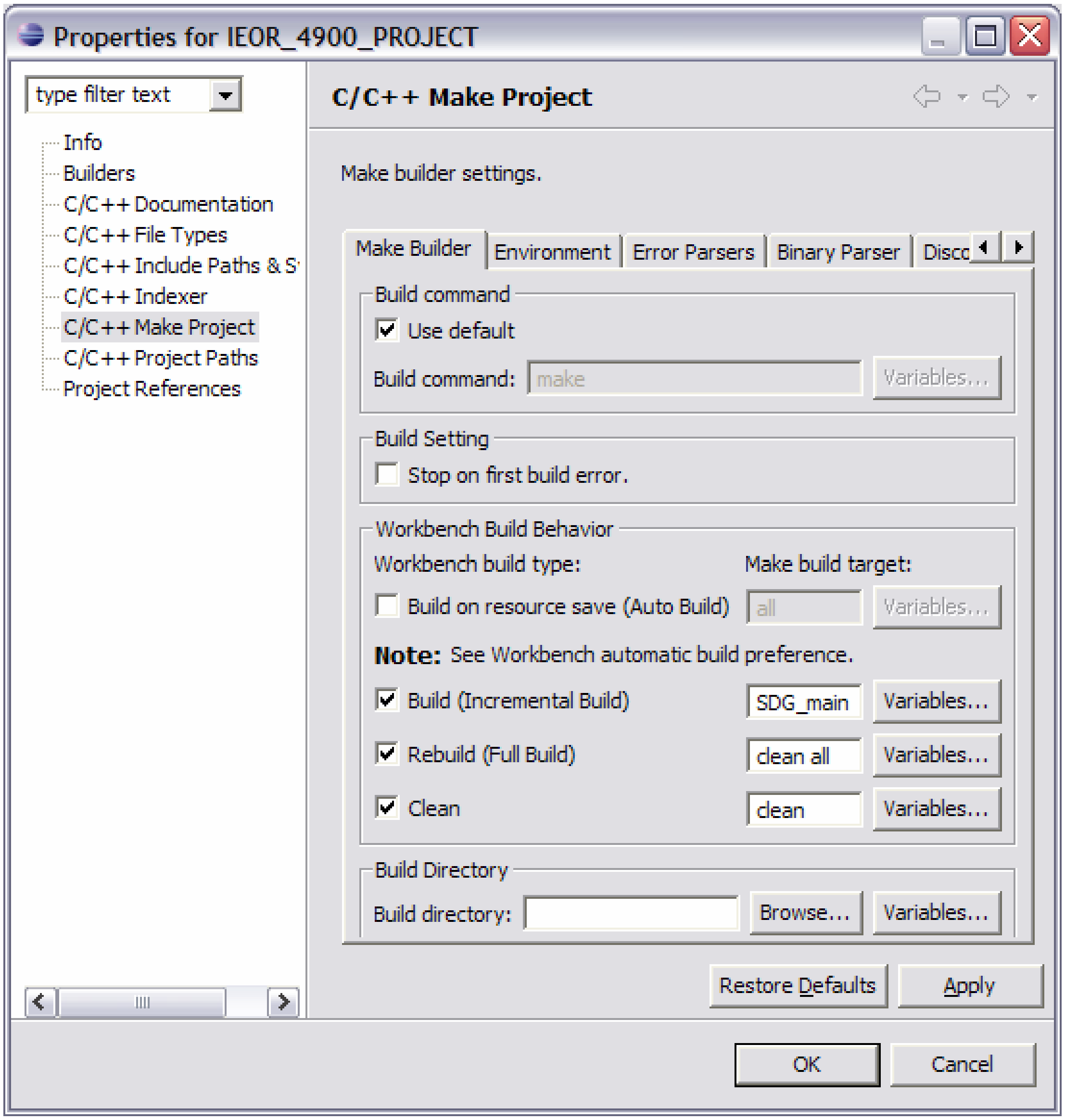
3. In the subsequent window, specify the project name, uncheck check box `Use default`, and specify the project directory. The project directory is the path under which all the `makefile`, `*.h`, and `*.cc` exist. Click `Next`.



4. In the following workbench window, make sure that `GNU Elf Parser` and `Cygwin PE Parser` are selected. Click `Next`.



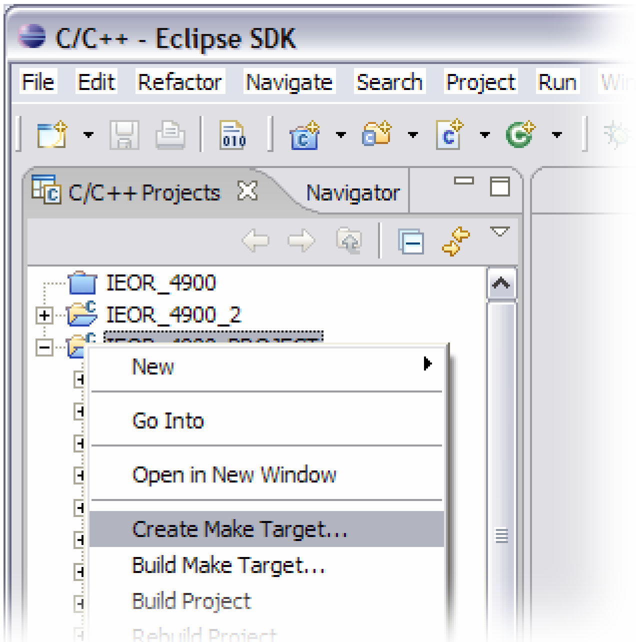
5. In the subsequent workbench window, specify `SDG_main` for testing the SDG mapping or `DGS_main` for testing the DGS mapping. The following screen snapshot shows how to define `SDG_main`. Click `OK`.



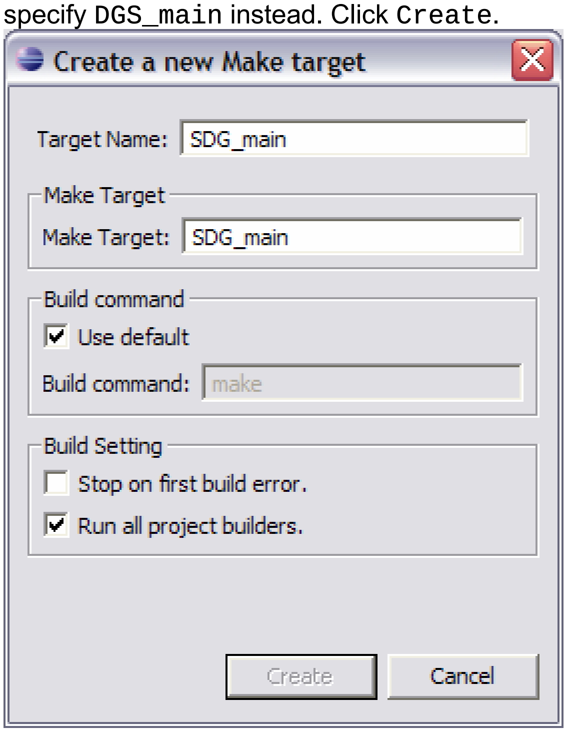
Compile

Follow the following steps to compile the application:

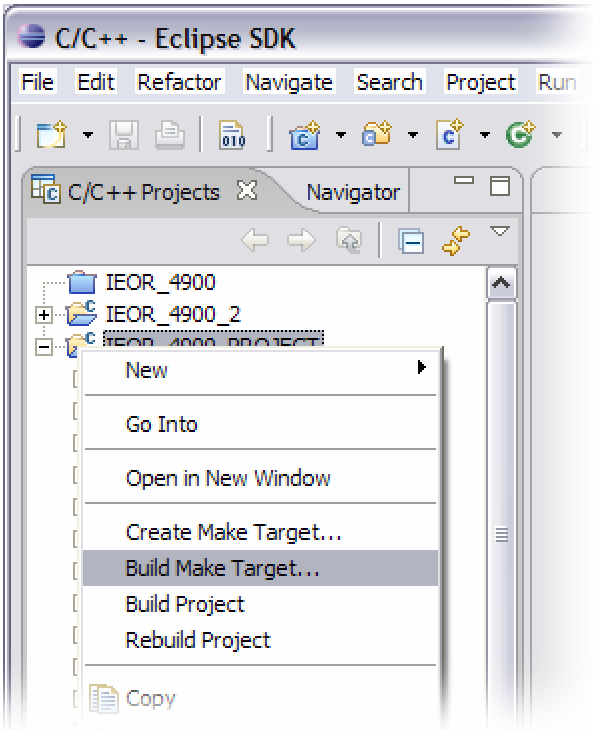
1. Right click on the name of the project that was just created. Select `Create Make Target...`.



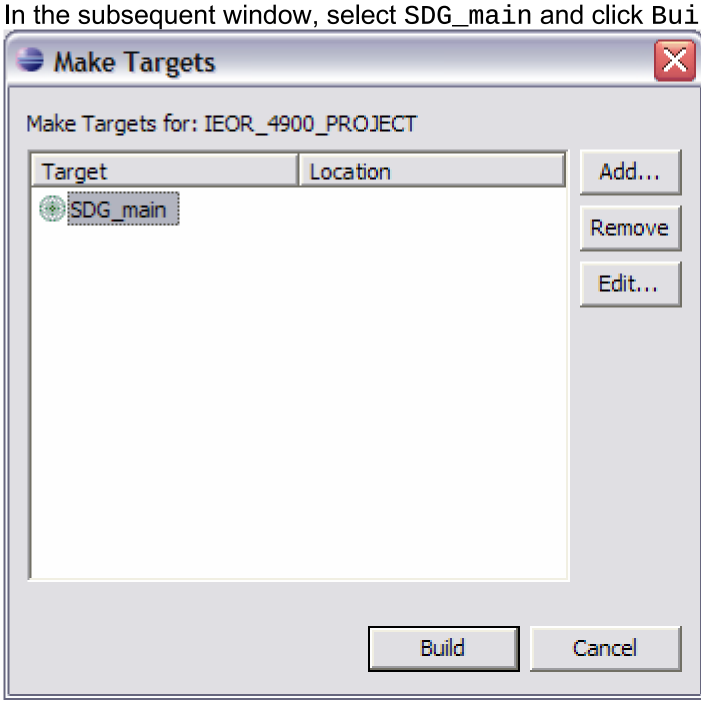
2. Specify `SDG_main` for `Target Name` and `Make Target`. In the case of testing DGS mapping, specify `DGS_main` instead. Click `Create`.



3. Right click on the name of the project and select `Build Make Target`....



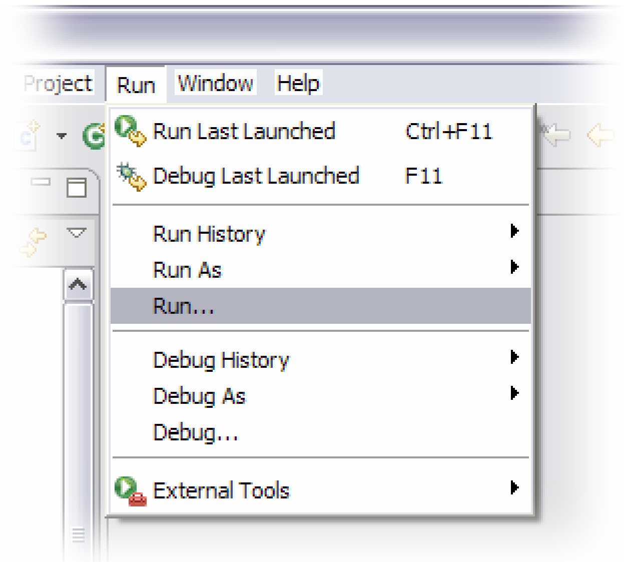
4. In the subsequent window, select `SDG_main` and click `Build`.



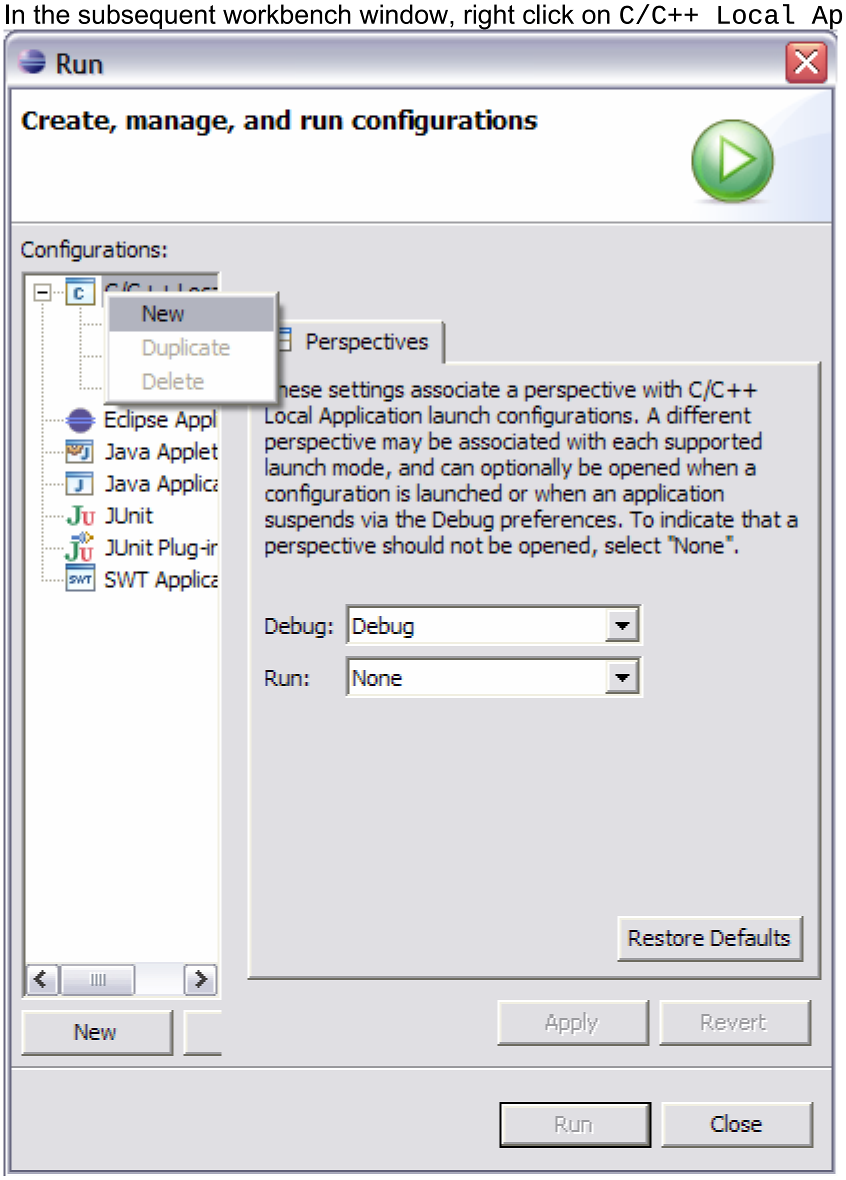
Test

Follow the following steps to compile the application:

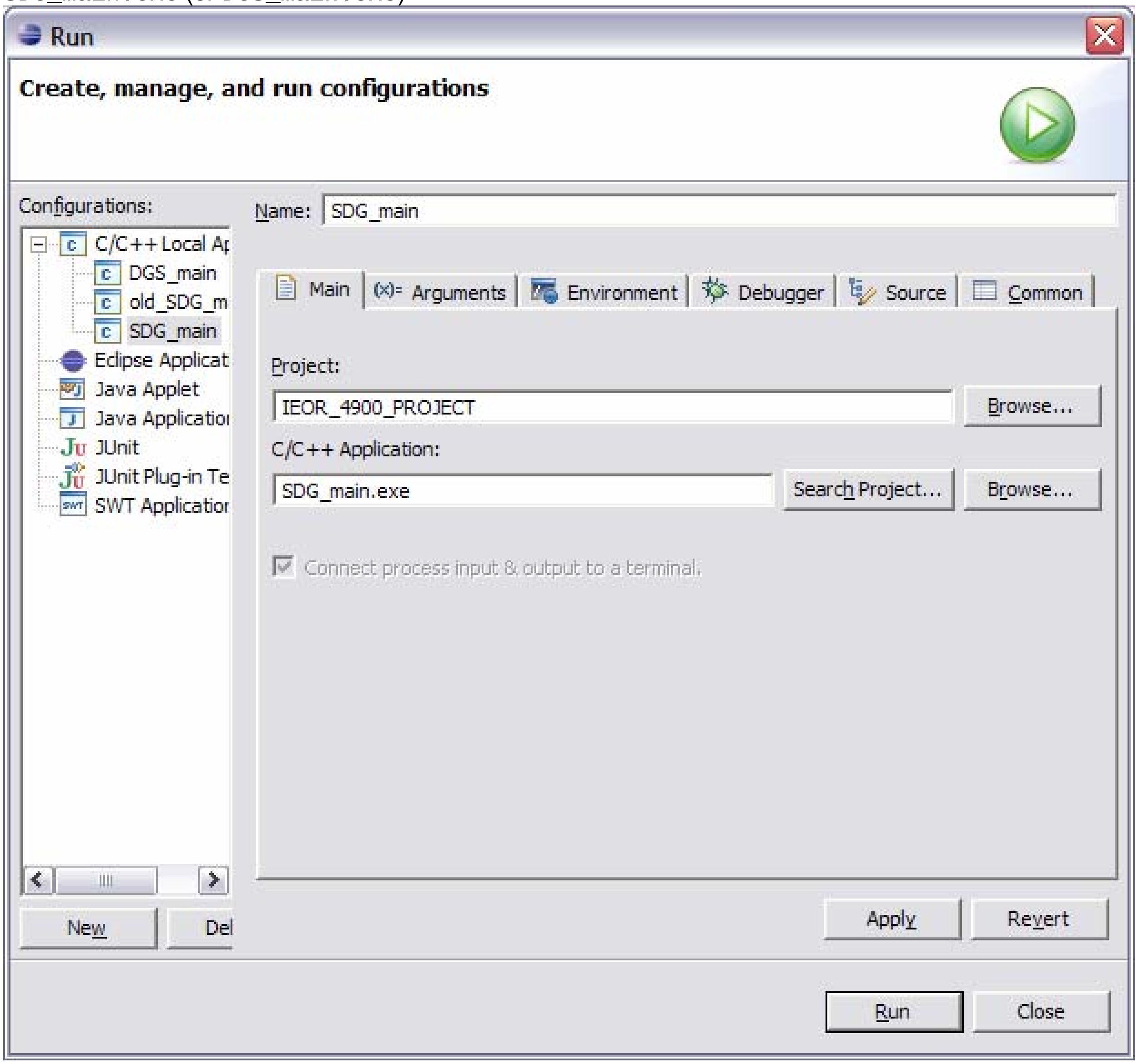
1. Retrieve the following files and copy them to the project path:
   a. `flows.dat`
   b. `nw.dat`
   c. `nw.pth`
   d. `odflows.dat`
   e. run control file (<test case name>`.run`>
2. In Eclipse, highlight the project name and then select menu `Run` > `Run...`.



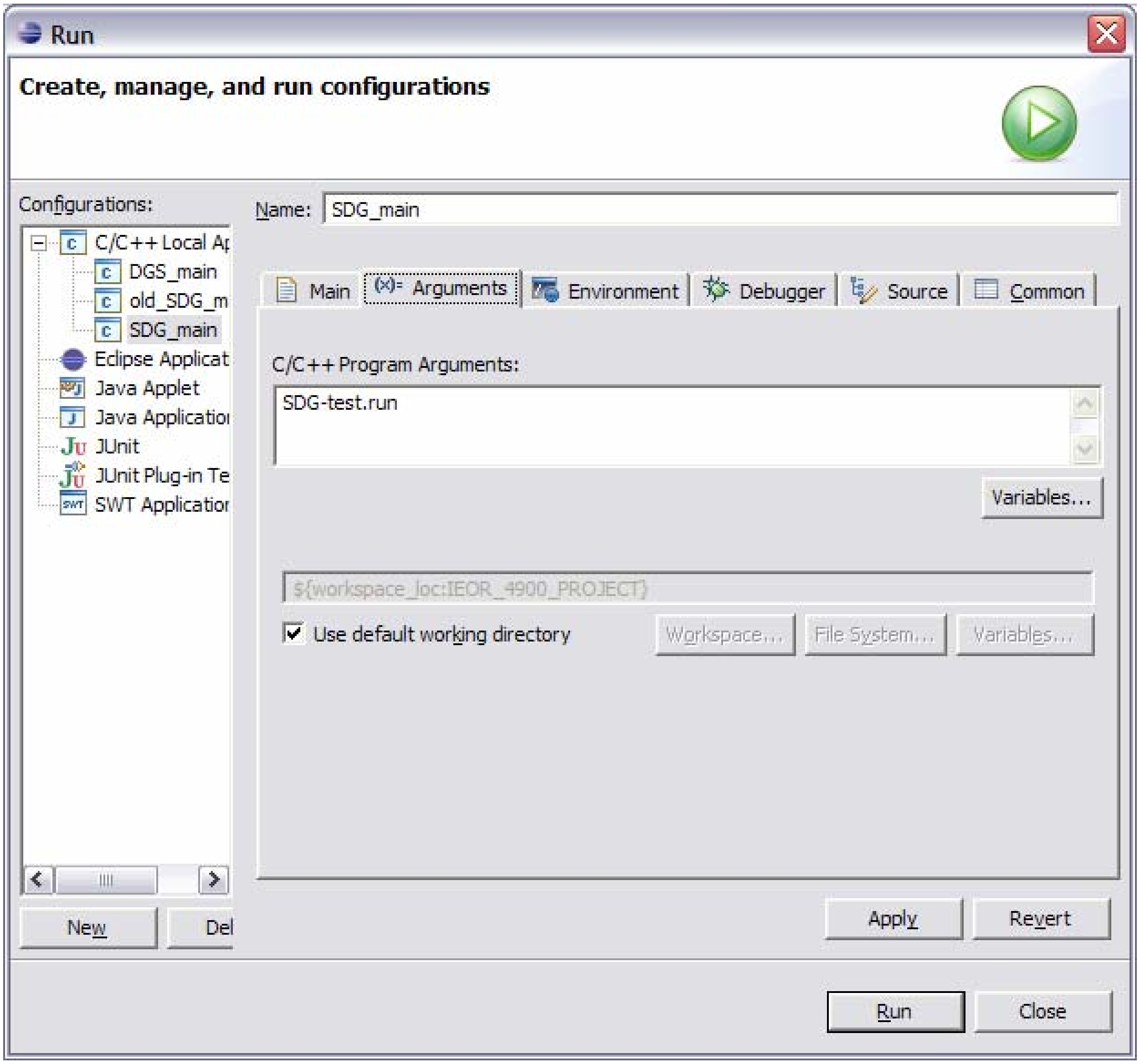
3. In the subsequent workbench window, right click on `C/C++ Local Application` and select `New`.



4. In the subsequent workbench window, make sure that `C/C++ Application` has the value `SDG_main.exe` (or `DGS_main.exe`)

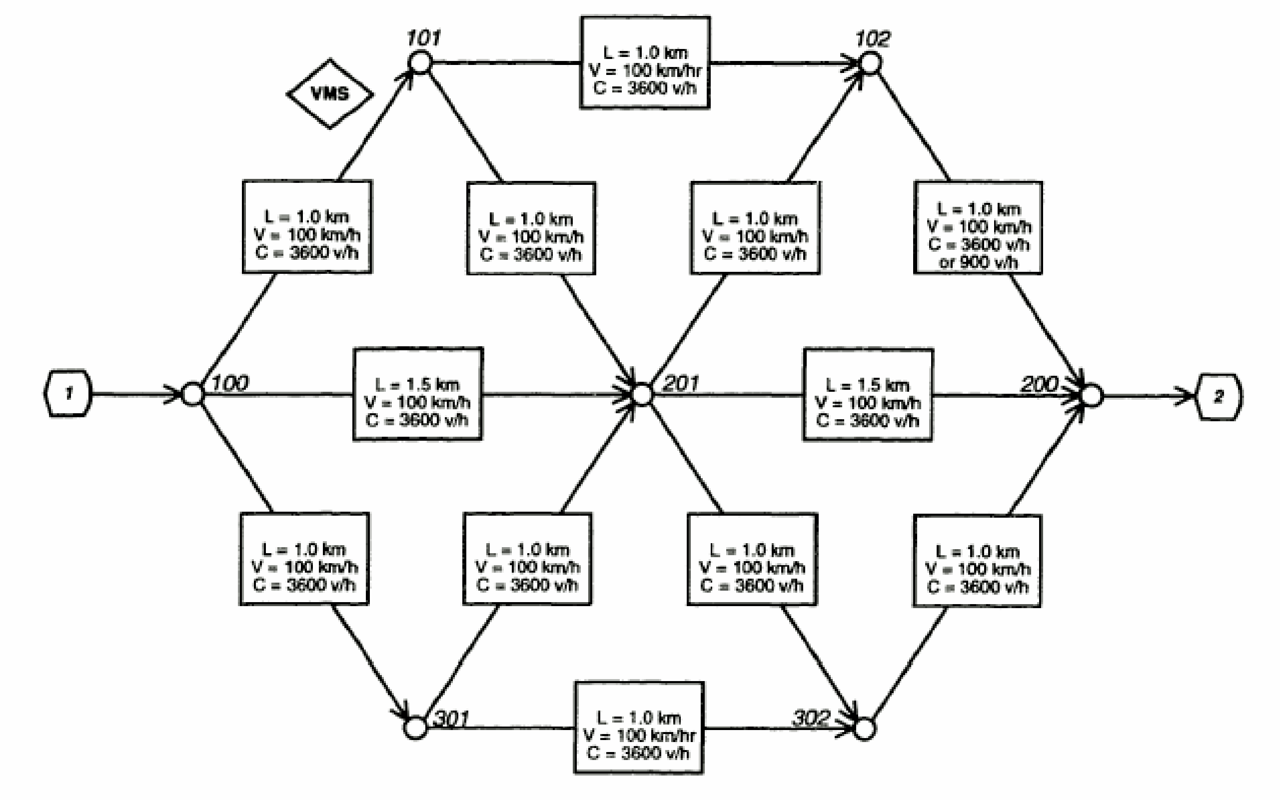


5. Select tab `Arguments` and type the name of the run control file (<test case name>`.run`). Click `Run`.



Description of the Test Network

Runs were made using a simple 14-link network with a single origin-destination (OD) pair and eleven OD paths. The following diagram shows this network. All links are single lane and 1 km long except for links 2 (100-201), which are 1.5 km long. Vehicles are assumed to be 7.5 m long, so the link storage capacities are about 133 and 200 vehicles, respectively. Free speed on all links is 100 km/hr. Except for the centroid connectors, all links normally have an exit capacity of 3,600 vehicles/ hour. In some runs, link 6 (102-200) has a reduced capacity of 900 vehicles/ hour throughout the entire simulation period. Both the origin and the destination centroid connectors (links 1-100 and 200-2, respectively) have infinite storage capacity (i.e, they do not spill back).


101
102
VMS
L = 1.0 km
V = 100 km/hr
C = 3600 v/h
L = 1.0 km
V = 100 km/h
C = 3600 v/h
L = 1.0 km
V = 100 km/h
C = 3600 v/h
L = 1.0 km
V = 100 km/h
C = 3600 v/h
L = 1.0 km
V = 100 km/h
C = 3600 v/h
or 900 v/h
1
100
L = 1.5 km
V = 100 km/h
C = 3600 v/h
201
L = 1.5 km
V = 100 km/h
C = 3600 v/h
200
2
L = 1.0 km
V = 100 km/h
C = 3600 v/h
L = 1.0 km
V = 100 km/h
C = 3600 v/h
L = 1.0 km
V = 100 km/h
C = 3600 v/h
L = 1.0 km
V = 100 km/h
C = 3600 v/h
301
L = 1.0 km
V = 100 km/hr
C = 3600 v/h
302

Graphic Outputs

Test Case Name	Network	Guidance	Path Splits	Flows	Times	Set of Code	Location in CD
`2_0_10000_ITERATIONS, SDG Mapping`	full capacity	100%	disaggregate	x1	x10	`CODE_2_0_ORIGINAL_MIGRATED`	<CD Root>`/CODE_2_0_ORIGINAL_MIGRATED`
`2_0_10000_ITERATIONS, DGS Mapping`	full capacity	100%	disaggregate	x1	x10	`CODE_2_0_ORIGINAL_MIGRATED`	<CD Root>`/CODE_2_0_ORIGINAL_MIGRATED`
`2_0_RELAXED_EXIT_CAPACITY, SDG Mapping`	full capacity	100%	disaggregate	x1	x10	`CODE_2_0_ORIGINAL_MIGRATED`	<CD Root>`/CODE_2_0_ORIGINAL_MIGRATED`
`2_0_RELAXED_EXIT_CAPACITY, DGS Mapping`	full capacity	100%	disaggregate	x1	x10	`CODE_2_0_ORIGINAL_MIGRATED`	<CD Root>`/CODE_2_0_ORIGINAL_MIGRATED`
`2_0_RELAXED_STORAGE_CAPACITY, SDG Mapping`	full capacity	100%	disaggregate	x1	x10	`CODE_2_0_ORIGINAL_MIGRATED`	<CD Root>`/CODE_2_0_ORIGINAL_MIGRATED`
`2_0_RELAXED_STORAGE_CAPACITY, DGS Mapping`	full capacity	100%	disaggregate	x1	x10	`CODE_2_0_ORIGINAL_MIGRATED`	<CD Root>`/CODE_2_0_ORIGINAL_MIGRATED`
`2_1_MSA_THEN_POLYAK`	full capacity	100%	disaggregate	x1	x10	`CODE_2_1_MSA_THEN_POLYAK`	<CD Root>`/CODE_2_1_MSA_THEN_POLYAK`

Test Case Name	Network	Guidance	Path Splits	Flows	Times	Set of Code	Location in CD
`2_2_POTENTIALS, Network2`	full capacity	100%	disaggregate	x1	x10	`CODE_2_2_POTENTIALS`	<CD Root>`/CODE_2_2_POTENTIALS`

$\min_{\alpha}\left\|mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right\|^2-\beta\alpha(1-\alpha)\left\|mapping\left(C^i\right)-C^i\right\|^2$, with multiplier (β) = 0, DGS Mapping

$\min_{\alpha}\left\|mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right\|^2-\beta\alpha(1-\alpha)\left\|mapping\left(C^i\right)-C^i\right\|^2$, with multiplier (β) = 0.001, DGS Mapping

$\min_{\alpha}\left\|mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right\|^2-\beta\alpha(1-\alpha)\left\|mapping\left(C^i\right)-C^i\right\|^2$, with multiplier (β) = 0.01, DGS Mapping

$\min_{\alpha}\left\|mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right\|^2-\beta\alpha(1-\alpha)\left\|mapping\left(C^i\right)-C^i\right\|^2$, with multiplier (β) = 0.1, DGS Mapping

$\min_{\alpha}\left\|mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right\|^2-\beta\alpha(1-\alpha)\left\|mapping\left(C^i\right)-C^i\right\|^2$, with multiplier (β) = 1, DGS Mapping

$\min_{\alpha}\left\|mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right\|^2-\beta\alpha^2\left\|mapping\left(C^i\right)-C^i\right\|^2$, with multiplier (β) = 0, DGS Mapping

$\min_{\alpha}\left\|mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right\|^2-\beta\alpha^2\left\|mapping\left(C^i\right)-C^i\right\|^2$, with multiplier (β) = 0.001, DGS Mapping

$\min_{\alpha}\left\|mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right\|^2-\beta\alpha^2\left\|mapping\left(C^i\right)-C^i\right\|^2$, with multiplier (β) = 0.01, DGS Mapping

$\min_{\alpha}\left\|mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right\|^2-\beta\alpha^2\left\|mapping\left(C^i\right)-C^i\right\|^2$, with multiplier (β) = 0.1, DGS Mapping

$\min_{\alpha}\left\|mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right\|^2-\beta\alpha^2\left\|mapping\left(C^i\right)-C^i\right\|^2$, with multiplier (β) = 1, DGS Mapping

$\min_{\alpha}\left\|mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right\|^2-\beta\alpha(1-\alpha)\left\|mapping\left(C^i\right)-C^i\right\|^2$, with multiplier (β) = 0, SDG Mapping

$\min_{\alpha}\left\|mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right\|^2-\beta\alpha(1-\alpha)\left\|mapping\left(C^i\right)-C^i\right\|^2$, with multiplier (β) = 0.001, SDG Mapping

$\min_{\alpha}\left\|mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right\|^2-\beta\alpha(1-\alpha)\left\|mapping\left(C^i\right)-C^i\right\|^2$, with multiplier (β) = 0.01, SDG Mapping

$\min_{\alpha}\left\|mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right\|^2-\beta\alpha(1-\alpha)\left\|mapping\left(C^i\right)-C^i\right\|^2$, with multiplier (β) = 0.1, SDG Mapping

$\min_{\alpha}\left\|mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right\|^2-\beta\alpha(1-\alpha)\left\|mapping\left(C^i\right)-C^i\right\|^2$, with multiplier (β) = 1, SDG Mapping

$$\min_{\alpha}\left\|mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right\|^2-\beta\alpha^2\left\|mapping\left(C^i\right)-C^i\right\|^2$$, with multiplier (β) = 0, SDG Mapping

$$\min_{\alpha}\left\|mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right\|^2-\beta\alpha^2\left\|mapping\left(C^i\right)-C^i\right\|^2$$, with multiplier (β) = 0.001, SDG Mapping

$$\min_{\alpha}\left\|mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right\|^2-\beta\alpha^2\left\|mapping\left(C^i\right)-C^i\right\|^2$$, with multiplier (β) = 0.01, SDG Mapping

$$\min_{\alpha}\left\|mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right\|^2-\beta\alpha^2\left\|mapping\left(C^i\right)-C^i\right\|^2$$, with multiplier (β) = 0.1, SDG Mapping

$$\min_{\alpha}\left\|mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right\|^2-\beta\alpha^2\left\|mapping\left(C^i\right)-C^i\right\|^2$$, with multiplier (β) = 1, SDG Mapping

$$\min_{\alpha}\left\|mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right\|-\beta\alpha(1-\alpha)\left\|mapping\left(C^i\right)-C^i\right\|$$, with multiplier (β) = 0, SDG Mapping

$$\min_{\alpha}\left\|mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right\|-\beta\alpha(1-\alpha)\left\|mapping\left(C^i\right)-C^i\right\|$$, with multiplier (β) = 0.001, SDG Mapping

$$\min_{\alpha}\left\|mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right\|-\beta\alpha(1-\alpha)\left\|mapping\left(C^i\right)-C^i\right\|$$, with multiplier (β) = 0.01, SDG Mapping

$$\min_{\alpha}\left\|mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right\|-\beta\alpha(1-\alpha)\left\|mapping\left(C^i\right)-C^i\right\|$$, with multiplier (β) = 0.1, SDG Mapping

$$\min_{\alpha}\left\|mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right\|-\beta\alpha(1-\alpha)\left\|mapping\left(C^i\right)-C^i\right\|$$, with multiplier (β) = 1, SDG Mapping

$$\min_{\alpha}\ln\left(\left\|mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right\|^2\right)-\beta\ln\left(\alpha(1-\alpha)\left\|mapping\left(C^i\right)-C^i\right\|^2\right)$$. with multiplier (β) = 0, SDG Mapping

$$\min_{\alpha}\ln\left(\left\|mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right\|^2\right)-\beta\ln\left(\alpha(1-\alpha)\left\|mapping\left(C^i\right)-C^i\right\|^2\right)$$. with multiplier (β) = 0.001, SDG Mapping

$$\min_{\alpha}\ln\left(\left\|mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right\|^2\right)-\beta\ln\left(\alpha(1-\alpha)\left\|mapping\left(C^i\right)-C^i\right\|^2\right)$$. with multiplier (β) = 0.01, SDG Mapping

$$\min_{\alpha}\ln\left(\left\|mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right\|^2\right)-\beta\ln\left(\alpha(1-\alpha)\left\|mapping\left(C^i\right)-C^i\right\|^2\right)$$. with multiplier (β) = 0.1, SDG Mapping

$$\min_{\alpha}\ln\left(\left\|mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right\|^2\right)-\beta\ln\left(\alpha(1-\alpha)\left\|mapping\left(C^i\right)-C^i\right\|^2\right)$$. with multiplier (β) = 1, SDG Mapping

$$\begin{aligned}&\min_{\alpha}\ln\left(\left\|mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right\|^2\right)\\&-\beta\ln\left(\alpha\left\|mapping\left(C^i\right)-C^i\right\|\right)\ln\left((1-\alpha)\left\|mapping\left(C^i\right)-C^i\right\|\right)\end{aligned}$$. with multiplier (β) = 0, SDG Mapping

$$\begin{aligned}&\min_{\alpha}\ln\left(\left\|mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right\|^2\right)\\&-\beta\ln\left(\alpha\left\|mapping\left(C^i\right)-C^i\right\|\right)\ln\left((1-\alpha)\left\|mapping\left(C^i\right)-C^i\right\|\right)\end{aligned}$$. with multiplier (β) = 0.001, SDG Mapping

$$\begin{aligned}&\min_{\alpha}\ln\left(\left\|mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right\|^2\right)\\&-\beta\ln\left(\alpha\left\|mapping\left(C^i\right)-C^i\right\|\right)\ln\left((1-\alpha)\left\|mapping\left(C^i\right)-C^i\right\|\right)\end{aligned}$$. with multiplier (β) = 0.01, SDG Mapping

$$\min_{\alpha} \ln\left(\left\|mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right\|^2\right) - \beta\ln\left(\alpha\left\|mapping\left(C^i\right)-C^i\right\|\right)\ln\left((1-\alpha)\left\|mapping\left(C^i\right)-C^i\right\|\right)$$. with multiplier (β) = 0.1, SDG Mapping

$$\min_{\alpha} \ln\left(\left\|mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right\|^2\right) - \beta\ln\left(\alpha\left\|mapping\left(C^i\right)-C^i\right\|\right)\ln\left((1-\alpha)\left\|mapping\left(C^i\right)-C^i\right\|\right)$$. with multiplier (β) = 1, SDG Mapping

$\min_{\alpha}\left(\left(mapping\left(C^i\right)-C^i\right)^t\left(mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right)\right)^2 - \beta\alpha(1-\alpha)\left\|mapping\left(C^i\right)-C^i\right\|^4$. with multiplier (β) = 0, SDG Mapping

$\min_{\alpha}\left(\left(mapping\left(C^i\right)-C^i\right)^t\left(mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right)\right)^2 - \beta\alpha(1-\alpha)\left\|mapping\left(C^i\right)-C^i\right\|^4$. with multiplier (β) = 0.001, SDG Mapping

$\min_{\alpha}\left(\left(mapping\left(C^i\right)-C^i\right)^t\left(mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right)\right)^2 - \beta\alpha(1-\alpha)\left\|mapping\left(C^i\right)-C^i\right\|^4$. with multiplier (β) = 0.01, SDG Mapping

$\min_{\alpha}\left(\left(mapping\left(C^i\right)-C^i\right)^t\left(mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right)\right)^2 - \beta\alpha(1-\alpha)\left\|mapping\left(C^i\right)-C^i\right\|^4$. with multiplier (β) = 0.1, SDG Mapping

$\min_{\alpha}\left(\left(mapping\left(C^i\right)-C^i\right)^t\left(mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right)\right)^2 - \beta\alpha(1-\alpha)\left\|mapping\left(C^i\right)-C^i\right\|^4$. with multiplier (β) = 1, SDG Mapping

$\min_{\alpha}\left(\left(mapping\left(C^i\right)-C^i\right)^t\left(mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right)\right)^2 - \beta\alpha^2\left\|mapping\left(C^i\right)-C^i\right\|^4$. with multiplier (β) = 0, SDG Mapping

$\min_{\alpha}\left(\left(mapping\left(C^i\right)-C^i\right)^t\left(mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right)\right)^2 - \beta\alpha^2\left\|mapping\left(C^i\right)-C^i\right\|^4$. with multiplier (β) = 0.001, SDG Mapping

$\min_{\alpha}\left(\left(mapping\left(C^i\right)-C^i\right)^t\left(mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right)\right)^2 - \beta\alpha^2\left\|mapping\left(C^i\right)-C^i\right\|^4$. with multiplier (β) = 0.01, SDG Mapping

$\min_{\alpha}\left(\left(mapping\left(C^i\right)-C^i\right)^t\left(mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right)\right)^2 - \beta\alpha^2\left\|mapping\left(C^i\right)-C^i\right\|^4$. with multiplier (β) = 0.1, SDG Mapping

$\min_{\alpha}\left(\left(mapping\left(C^i\right)-C^i\right)^t\left(mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right)\right)^2 - \beta\alpha^2\left\|mapping\left(C^i\right)-C^i\right\|^4$. with multiplier (β) = 1, SDG Mapping

$\min_{\alpha}\left(mapping\left(C^i\right)-C^i\right)^t\left(mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right) - \beta\alpha(1-\alpha)\left\|mapping\left(C^i\right)-C^i\right\|^2$. with multiplier (β) = 0, SDG Mapping

$\min_{\alpha}\left(mapping\left(C^i\right)-C^i\right)^t\left(mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right) - \beta\alpha(1-\alpha)\left\|mapping\left(C^i\right)-C^i\right\|^2$. with multiplier (β) = 0.001, SDG Mapping

$\min_{\alpha}\left(mapping\left(C^i\right)-C^i\right)^t\left(mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right) - \beta\alpha(1-\alpha)\left\|mapping\left(C^i\right)-C^i\right\|^2$. with multiplier (β) = 0.01, SDG Mapping

$\min_{\alpha}\left(mapping\left(C^i\right)-C^i\right)^t\left(mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right) - \beta\alpha(1-\alpha)\left\|mapping\left(C^i\right)-C^i\right\|^2$. with multiplier (β) = 0.1, SDG Mapping

$\min_{\alpha}\left(mapping\left(C^i\right)-C^i\right)^t\left(mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right) - \beta\alpha(1-\alpha)\left\|mapping\left(C^i\right)-C^i\right\|^2$. with multiplier (β) = 1, SDG Mapping

$\min_{\alpha}\ln\left(\left(\left(mapping\left(C^i\right)-C^i\right)^t\left(mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right)\right)^2\right) - \beta\ln\left(\alpha(1-\alpha)\left\|mapping\left(C^i\right)-C^i\right\|^4\right)$. with multiplier (β) = 0, SDG Mapping

$$\min_{\alpha} \ln\left(\left(\left(mapping\left(C^i\right)-C^i\right)^t\left(mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right)\right)^2\right)-\beta\ln\left(\alpha(1-\alpha)\left\|mapping\left(C^i\right)-C^i\right\|^4\right)$$
. with multiplier (β) = 0.001, SDG Mapping

$$\min_{\alpha} \ln\left(\left(\left(mapping\left(C^i\right)-C^i\right)^t\left(mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right)\right)^2\right)-\beta\ln\left(\alpha(1-\alpha)\left\|mapping\left(C^i\right)-C^i\right\|^4\right)$$
. with multiplier (β) = 0.01, SDG Mapping

$$\min_{\alpha} \ln\left(\left(\left(mapping\left(C^i\right)-C^i\right)^t\left(mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right)\right)^2\right)-\beta\ln\left(\alpha(1-\alpha)\left\|mapping\left(C^i\right)-C^i\right\|^4\right)$$
. with multiplier (β) = 0.1, SDG Mapping

$$\min_{\alpha} \ln\left(\left(\left(mapping\left(C^i\right)-C^i\right)^t\left(mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right)\right)^2\right)-\beta\ln\left(\alpha(1-\alpha)\left\|mapping\left(C^i\right)-C^i\right\|^4\right)$$
. with multiplier (β) = 0, SDG Mapping

$$\min_{\alpha} \ln\left(\left(\left(mapping\left(C^i\right)-C^i\right)^t\left(mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right)\right)^2\right)$$
$$-\beta\ln\left(\alpha\left\|mapping\left(C^i\right)-C^i\right\|^2\right)\ln\left((1-\alpha)\left\|mapping\left(C^i\right)-C^i\right\|^2\right)$$
. with multiplier (β) = 0, SDG Mapping

$$\min_{\alpha} \ln\left(\left(\left(mapping\left(C^i\right)-C^i\right)^t\left(mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right)\right)^2\right)$$
$$-\beta\ln\left(\alpha\left\|mapping\left(C^i\right)-C^i\right\|^2\right)\ln\left((1-\alpha)\left\|mapping\left(C^i\right)-C^i\right\|^2\right)$$
. with multiplier (β) = 0.001, SDG Mapping

$$\min_{\alpha} \ln\left(\left(\left(mapping\left(C^i\right)-C^i\right)^t\left(mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right)\right)^2\right)$$
$$-\beta\ln\left(\alpha\left\|mapping\left(C^i\right)-C^i\right\|^2\right)\ln\left((1-\alpha)\left\|mapping\left(C^i\right)-C^i\right\|^2\right)$$
. with multiplier (β) = 0.01, SDG Mapping

$$\min_{\alpha} \ln\left(\left(\left(mapping\left(C^i\right)-C^i\right)^t\left(mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right)\right)^2\right)$$
$$-\beta\ln\left(\alpha\left\|mapping\left(C^i\right)-C^i\right\|^2\right)\ln\left((1-\alpha)\left\|mapping\left(C^i\right)-C^i\right\|^2\right)$$
. with multiplier (β) = 0.1, SDG Mapping

$$\min_{\alpha} \ln\left(\left(\left(mapping\left(C^i\right)-C^i\right)^t\left(mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right)\right)^2\right)$$
$$-\beta\ln\left(\alpha\left\|mapping\left(C^i\right)-C^i\right\|^2\right)\ln\left((1-\alpha)\left\|mapping\left(C^i\right)-C^i\right\|^2\right)$$
. with multiplier (β) = 1, SDG Mapping

Test Case Name	Network	Guidance	Path Splits	Flows	Times	Set of Code	Location in CD
`2_2_POTENTIALS, Network2, smaller`	full capacity smaller	100%	disaggregate	x1	x10	`CODE_2_2_POTENTIALS`	<CD Root>/`CODE_2_2_POTENTIALS`

$$\min_{\alpha}\left\|mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right\|^2-\beta\alpha(1-\alpha)\left\|mapping\left(C^i\right)-C^i\right\|^2$$
, with multiplier (β) = 0, DGS Mapping

$$\min_{\alpha}\left\|mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right\|^2-\beta\alpha(1-\alpha)\left\|mapping\left(C^i\right)-C^i\right\|^2$$
, with multiplier (β) = 0.001, DGS Mapping

$\min_{\alpha}\left\|mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right\|^2-\beta\alpha(1-\alpha)\left\|mapping\left(C^i\right)-C^i\right\|^2$, with multiplier (β) = 0.01, DGS Mapping

$\min_{\alpha}\left\|mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right\|^2-\beta\alpha(1-\alpha)\left\|mapping\left(C^i\right)-C^i\right\|^2$, with multiplier (β) = 0.1, DGS Mapping

$\min_{\alpha}\left\|mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right\|^2-\beta\alpha(1-\alpha)\left\|mapping\left(C^i\right)-C^i\right\|^2$, with multiplier (β) = 1, DGS Mapping

$\min_{\alpha}\left\|mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right\|^2-\beta\alpha^2\left\|mapping\left(C^i\right)-C^i\right\|^2$, with multiplier (β) = 0, DGS Mapping

$\min_{\alpha}\left\|mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right\|^2-\beta\alpha^2\left\|mapping\left(C^i\right)-C^i\right\|^2$, with multiplier (β) = 0.001, DGS Mapping

$\min_{\alpha}\left\|mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right\|^2-\beta\alpha^2\left\|mapping\left(C^i\right)-C^i\right\|^2$, with multiplier (β) = 0.01, DGS Mapping

$\min_{\alpha}\left\|mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right\|^2-\beta\alpha^2\left\|mapping\left(C^i\right)-C^i\right\|^2$, with multiplier (β) = 0.1, DGS Mapping

$\min_{\alpha}\left\|mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right\|^2-\beta\alpha^2\left\|mapping\left(C^i\right)-C^i\right\|^2$, with multiplier (β) = 1, DGS Mapping

$\min_{\alpha}\left\|mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right\|^2-\beta\alpha(1-\alpha)\left\|mapping\left(C^i\right)-C^i\right\|^2$, with multiplier (β) = 0, SDG Mapping

$\min_{\alpha}\left\|mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right\|^2-\beta\alpha(1-\alpha)\left\|mapping\left(C^i\right)-C^i\right\|^2$, with multiplier (β) = 0.001, SDG Mapping

$\min_{\alpha}\left\|mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right\|^2-\beta\alpha(1-\alpha)\left\|mapping\left(C^i\right)-C^i\right\|^2$, with multiplier (β) = 0.01, SDG Mapping

$\min_{\alpha}\left\|mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right\|^2-\beta\alpha(1-\alpha)\left\|mapping\left(C^i\right)-C^i\right\|^2$, with multiplier (β) = 0.1, SDG Mapping

$\min_{\alpha}\left\|mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right\|^2-\beta\alpha(1-\alpha)\left\|mapping\left(C^i\right)-C^i\right\|^2$, with multiplier (β) = 1, SDG Mapping

$\min_{\alpha}\left\|mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right\|^2-\beta\alpha^2\left\|mapping\left(C^i\right)-C^i\right\|^2$, with multiplier (β) = 0, SDG Mapping

$\min_{\alpha}\left\|mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right\|^2-\beta\alpha^2\left\|mapping\left(C^i\right)-C^i\right\|^2$, with multiplier (β) = 0.001, SDG Mapping

$\min_{\alpha}\left\|mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right\|^2-\beta\alpha^2\left\|mapping\left(C^i\right)-C^i\right\|^2$, with multiplier (β) = 0.01, SDG Mapping

$\min_{\alpha}\left\|mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right\|^2-\beta\alpha^2\left\|mapping\left(C^i\right)-C^i\right\|^2$, with multiplier (β) = 0.1, SDG Mapping

$\min_{\alpha}\left\|mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right\|^2-\beta\alpha^2\left\|mapping\left(C^i\right)-C^i\right\|^2$, with multiplier (β) = 1, SDG Mapping

Test Case Name	Network	Guidance	Path Splits	Flows	Times	Set of Code	Location in CD
`2_2_POTENTIALS, Network3`	link 102 -200 at 900 v/h	100%	Aggregate	x10	x10	`CODE_2_2_POTENTIALS`	<CD Root>`/CODE_2_2_POTENTIALS`

$\min_{\alpha}\left\|mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right\|^2-\beta\alpha(1-\alpha)\left\|mapping\left(C^i\right)-C^i\right\|^2$, with multiplier (β) = 0, DGS Mapping

$\min_{\alpha}\left\|mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right\|^2-\beta\alpha(1-\alpha)\left\|mapping\left(C^i\right)-C^i\right\|^2$, with multiplier (β) = 0.001, DGS Mapping

$\min_{\alpha}\left\|mapping\left(C^{i}(\alpha)\right)-C^{i}(\alpha)\right\|^{2}-\beta\alpha(1-\alpha)\left\|mapping\left(C^{i}\right)-C^{i}\right\|^{2}$, with multiplier (β) = 0.01, DGS Mapping

$\min_{\alpha}\left\|mapping\left(C^{i}(\alpha)\right)-C^{i}(\alpha)\right\|^{2}-\beta\alpha(1-\alpha)\left\|mapping\left(C^{i}\right)-C^{i}\right\|^{2}$, with multiplier (β) = 0.1, DGS Mapping

$\min_{\alpha}\left\|mapping\left(C^{i}(\alpha)\right)-C^{i}(\alpha)\right\|^{2}-\beta\alpha(1-\alpha)\left\|mapping\left(C^{i}\right)-C^{i}\right\|^{2}$, with multiplier (β) = 1, DGS Mapping

$\min_{\alpha}\left\|mapping\left(C^{i}(\alpha)\right)-C^{i}(\alpha)\right\|^{2}-\beta\alpha^{2}\left\|mapping\left(C^{i}\right)-C^{i}\right\|^{2}$, with multiplier (β) = 0, DGS Mapping

$\min_{\alpha}\left\|mapping\left(C^{i}(\alpha)\right)-C^{i}(\alpha)\right\|^{2}-\beta\alpha^{2}\left\|mapping\left(C^{i}\right)-C^{i}\right\|^{2}$, with multiplier (β) = 0.001, DGS Mapping

$\min_{\alpha}\left\|mapping\left(C^{i}(\alpha)\right)-C^{i}(\alpha)\right\|^{2}-\beta\alpha^{2}\left\|mapping\left(C^{i}\right)-C^{i}\right\|^{2}$, with multiplier (β) = 0.01, DGS Mapping

$\min_{\alpha}\left\|mapping\left(C^{i}(\alpha)\right)-C^{i}(\alpha)\right\|^{2}-\beta\alpha^{2}\left\|mapping\left(C^{i}\right)-C^{i}\right\|^{2}$, with multiplier (β) = 0.1, DGS Mapping

$\min_{\alpha}\left\|mapping\left(C^{i}(\alpha)\right)-C^{i}(\alpha)\right\|^{2}-\beta\alpha^{2}\left\|mapping\left(C^{i}\right)-C^{i}\right\|^{2}$, with multiplier (β) = 1, DGS Mapping

$\min_{\alpha}\left\|mapping\left(C^{i}(\alpha)\right)-C^{i}(\alpha)\right\|^{2}-\beta\alpha(1-\alpha)\left\|mapping\left(C^{i}\right)-C^{i}\right\|^{2}$, with multiplier (β) = 0, SDG Mapping

$\min_{\alpha}\left\|mapping\left(C^{i}(\alpha)\right)-C^{i}(\alpha)\right\|^{2}-\beta\alpha(1-\alpha)\left\|mapping\left(C^{i}\right)-C^{i}\right\|^{2}$, with multiplier (β) = 0.001, SDG Mapping

$\min_{\alpha}\left\|mapping\left(C^{i}(\alpha)\right)-C^{i}(\alpha)\right\|^{2}-\beta\alpha(1-\alpha)\left\|mapping\left(C^{i}\right)-C^{i}\right\|^{2}$, with multiplier (β) = 0.01, SDG Mapping

$\min_{\alpha}\left\|mapping\left(C^{i}(\alpha)\right)-C^{i}(\alpha)\right\|^{2}-\beta\alpha(1-\alpha)\left\|mapping\left(C^{i}\right)-C^{i}\right\|^{2}$, with multiplier (β) = 0.1, SDG Mapping

$\min_{\alpha}\left\|mapping\left(C^{i}(\alpha)\right)-C^{i}(\alpha)\right\|^{2}-\beta\alpha(1-\alpha)\left\|mapping\left(C^{i}\right)-C^{i}\right\|^{2}$, with multiplier (β) = 1, SDG Mapping

$\min_{\alpha}\left\|mapping\left(C^{i}(\alpha)\right)-C^{i}(\alpha)\right\|^{2}-\beta\alpha^{2}\left\|mapping\left(C^{i}\right)-C^{i}\right\|^{2}$, with multiplier (β) = 0, SDG Mapping

$\min_{\alpha}\left\|mapping\left(C^{i}(\alpha)\right)-C^{i}(\alpha)\right\|^{2}-\beta\alpha^{2}\left\|mapping\left(C^{i}\right)-C^{i}\right\|^{2}$, with multiplier (β) = 0.001, SDG Mapping

$\min_{\alpha}\left\|mapping\left(C^{i}(\alpha)\right)-C^{i}(\alpha)\right\|^{2}-\beta\alpha^{2}\left\|mapping\left(C^{i}\right)-C^{i}\right\|^{2}$, with multiplier (β) = 0.01, SDG Mapping

$\min_{\alpha}\left\|mapping\left(C^{i}(\alpha)\right)-C^{i}(\alpha)\right\|^{2}-\beta\alpha^{2}\left\|mapping\left(C^{i}\right)-C^{i}\right\|^{2}$, with multiplier (β) = 0.1, SDG Mapping

$\min_{\alpha}\left\|mapping\left(C^{i}(\alpha)\right)-C^{i}(\alpha)\right\|^{2}-\beta\alpha^{2}\left\|mapping\left(C^{i}\right)-C^{i}\right\|^{2}$, with multiplier (β) = 1, SDG Mapping

$\min_{\alpha}\left\|mapping\left(C^{i}(\alpha)\right)-C^{i}(\alpha)\right\|-\beta\alpha(1-\alpha)\left\|mapping\left(C^{i}\right)-C^{i}\right\|$, with multiplier (β) = 0, SDG Mapping

$\min_{\alpha}\left\|mapping\left(C^{i}(\alpha)\right)-C^{i}(\alpha)\right\|-\beta\alpha(1-\alpha)\left\|mapping\left(C^{i}\right)-C^{i}\right\|$, with multiplier (β) = 0.001, SDG Mapping

$\min_{\alpha}\left\|mapping\left(C^{i}(\alpha)\right)-C^{i}(\alpha)\right\|-\beta\alpha(1-\alpha)\left\|mapping\left(C^{i}\right)-C^{i}\right\|$, with multiplier (β) = 0.01, SDG Mapping

$\min_{\alpha}\left\|mapping\left(C^{i}(\alpha)\right)-C^{i}(\alpha)\right\|-\beta\alpha(1-\alpha)\left\|mapping\left(C^{i}\right)-C^{i}\right\|$, with multiplier (β) = 0.1, SDG Mapping

$$\min_{\alpha}\left\|mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right\| - \beta\alpha(1-\alpha)\left\|mapping\left(C^i\right)-C^i\right\|$$, with multiplier (β) = 1, SDG Mapping

$$\min_{\alpha}\ln\left(\left\|mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right\|^2\right) - \beta\ln\left(\alpha(1-\alpha)\left\|mapping\left(C^i\right)-C^i\right\|^2\right)$$. with multiplier (β) = 0, SDG Mapping

$$\min_{\alpha}\ln\left(\left\|mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right\|^2\right) - \beta\ln\left(\alpha(1-\alpha)\left\|mapping\left(C^i\right)-C^i\right\|^2\right)$$. with multiplier (β) = 0.001, SDG Mapping

$$\min_{\alpha}\ln\left(\left\|mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right\|^2\right) - \beta\ln\left(\alpha(1-\alpha)\left\|mapping\left(C^i\right)-C^i\right\|^2\right)$$. with multiplier (β) = 0.01, SDG Mapping

$$\min_{\alpha}\ln\left(\left\|mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right\|^2\right) - \beta\ln\left(\alpha(1-\alpha)\left\|mapping\left(C^i\right)-C^i\right\|^2\right)$$. with multiplier (β) = 0.1, SDG Mapping

$$\min_{\alpha}\ln\left(\left\|mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right\|^2\right) - \beta\ln\left(\alpha(1-\alpha)\left\|mapping\left(C^i\right)-C^i\right\|^2\right)$$. with multiplier (β) = 1, SDG Mapping

$$\min_{\alpha}\ln\left(\left\|mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right\|^2\right) - \beta\ln\left(\alpha\left\|mapping\left(C^i\right)-C^i\right\|\right)\ln\left((1-\alpha)\left\|mapping\left(C^i\right)-C^i\right\|\right)$$. with multiplier (β) = 0, SDG Mapping

$$\min_{\alpha}\ln\left(\left\|mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right\|^2\right) - \beta\ln\left(\alpha\left\|mapping\left(C^i\right)-C^i\right\|\right)\ln\left((1-\alpha)\left\|mapping\left(C^i\right)-C^i\right\|\right)$$. with multiplier (β) = 0.001, SDG Mapping

$$\min_{\alpha}\ln\left(\left\|mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right\|^2\right) - \beta\ln\left(\alpha\left\|mapping\left(C^i\right)-C^i\right\|\right)\ln\left((1-\alpha)\left\|mapping\left(C^i\right)-C^i\right\|\right)$$. with multiplier (β) = 0.01, SDG Mapping

$$\min_{\alpha}\ln\left(\left\|mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right\|^2\right) - \beta\ln\left(\alpha\left\|mapping\left(C^i\right)-C^i\right\|\right)\ln\left((1-\alpha)\left\|mapping\left(C^i\right)-C^i\right\|\right)$$. with multiplier (β) = 0.1, SDG Mapping

$$\min_{\alpha}\ln\left(\left\|mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right\|^2\right) - \beta\ln\left(\alpha\left\|mapping\left(C^i\right)-C^i\right\|\right)\ln\left((1-\alpha)\left\|mapping\left(C^i\right)-C^i\right\|\right)$$. with multiplier (β) = 1, SDG Mapping

$$\min_{\alpha}\left(\left(mapping\left(C^i\right)-C^i\right)^t\left(mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right)\right)^2 - \beta\alpha(1-\alpha)\left\|mapping\left(C^i\right)-C^i\right\|^4$$. with multiplier (β) = 0, SDG Mapping

$$\min_{\alpha}\left(\left(mapping\left(C^i\right)-C^i\right)^t\left(mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right)\right)^2 - \beta\alpha(1-\alpha)\left\|mapping\left(C^i\right)-C^i\right\|^4$$. with multiplier (β) = 0.001, SDG Mapping

$$\min_{\alpha}\left(\left(mapping\left(C^i\right)-C^i\right)^t\left(mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right)\right)^2 - \beta\alpha(1-\alpha)\left\|mapping\left(C^i\right)-C^i\right\|^4$$. with multiplier (β) = 0.01, SDG Mapping

$$\min_{\alpha}\left(\left(mapping\left(C^i\right)-C^i\right)^t\left(mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right)\right)^2 - \beta\alpha(1-\alpha)\left\|mapping\left(C^i\right)-C^i\right\|^4$$. with multiplier (β) = 0.1, SDG Mapping

$$\min_{\alpha}\left(\left(mapping\left(C^i\right)-C^i\right)^t\left(mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right)\right)^2 - \beta\alpha(1-\alpha)\left\|mapping\left(C^i\right)-C^i\right\|^4$$. with multiplier (β) = 1, SDG Mapping

$\min_{\alpha}\left(\left(mapping\left(C^i\right)-C^i\right)^t\left(mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right)\right)^2-\beta\alpha^2\left\|mapping\left(C^i\right)-C^i\right\|^4$. with multiplier (β) = 0, SDG Mapping

$\min_{\alpha}\left(\left(mapping\left(C^i\right)-C^i\right)^t\left(mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right)\right)^2-\beta\alpha^2\left\|mapping\left(C^i\right)-C^i\right\|^4$. with multiplier (β) = 0.001, SDG Mapping

$\min_{\alpha}\left(\left(mapping\left(C^i\right)-C^i\right)^t\left(mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right)\right)^2-\beta\alpha^2\left\|mapping\left(C^i\right)-C^i\right\|^4$. with multiplier (β) = 0.01, SDG Mapping

$\min_{\alpha}\left(\left(mapping\left(C^i\right)-C^i\right)^t\left(mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right)\right)^2-\beta\alpha^2\left\|mapping\left(C^i\right)-C^i\right\|^4$. with multiplier (β) = 0.1, SDG Mapping

$\min_{\alpha}\left(\left(mapping\left(C^i\right)-C^i\right)^t\left(mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right)\right)^2-\beta\alpha^2\left\|mapping\left(C^i\right)-C^i\right\|^4$. with multiplier (β) = 1, SDG Mapping

$\min_{\alpha}\left(mapping\left(C^i\right)-C^i\right)^t\left(mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right)-\beta\alpha(1-\alpha)\left\|mapping\left(C^i\right)-C^i\right\|^2$. with multiplier (β) = 0, SDG Mapping

$\min_{\alpha}\left(mapping\left(C^i\right)-C^i\right)^t\left(mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right)-\beta\alpha(1-\alpha)\left\|mapping\left(C^i\right)-C^i\right\|^2$. with multiplier (β) = 0.001, SDG Mapping

$\min_{\alpha}\left(mapping\left(C^i\right)-C^i\right)^t\left(mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right)-\beta\alpha(1-\alpha)\left\|mapping\left(C^i\right)-C^i\right\|^2$. with multiplier (β) = 0.01, SDG Mapping

$\min_{\alpha}\left(mapping\left(C^i\right)-C^i\right)^t\left(mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right)-\beta\alpha(1-\alpha)\left\|mapping\left(C^i\right)-C^i\right\|^2$. with multiplier (β) = 0.1, SDG Mapping

$\min_{\alpha}\left(mapping\left(C^i\right)-C^i\right)^t\left(mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right)-\beta\alpha(1-\alpha)\left\|mapping\left(C^i\right)-C^i\right\|^2$. with multiplier (β) = 1, SDG Mapping

$\min_{\alpha}\ln\left(\left(\left(mapping\left(C^i\right)-C^i\right)^t\left(mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right)\right)^2\right)-\beta\ln\left(\alpha(1-\alpha)\left\|mapping\left(C^i\right)-C^i\right\|^4\right)$. with multiplier (β) = 0, SDG Mapping

$\min_{\alpha}\ln\left(\left(\left(mapping\left(C^i\right)-C^i\right)^t\left(mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right)\right)^2\right)-\beta\ln\left(\alpha(1-\alpha)\left\|mapping\left(C^i\right)-C^i\right\|^4\right)$. with multiplier (β) = 0.001, SDG Mapping

$\min_{\alpha}\ln\left(\left(\left(mapping\left(C^i\right)-C^i\right)^t\left(mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right)\right)^2\right)-\beta\ln\left(\alpha(1-\alpha)\left\|mapping\left(C^i\right)-C^i\right\|^4\right)$. with multiplier (β) = 0.01, SDG Mapping

$\min_{\alpha}\ln\left(\left(\left(mapping\left(C^i\right)-C^i\right)^t\left(mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right)\right)^2\right)-\beta\ln\left(\alpha(1-\alpha)\left\|mapping\left(C^i\right)-C^i\right\|^4\right)$. with multiplier (β) = 0.1, SDG Mapping

$\min_{\alpha}\ln\left(\left(\left(mapping\left(C^i\right)-C^i\right)^t\left(mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right)\right)^2\right)-\beta\ln\left(\alpha(1-\alpha)\left\|mapping\left(C^i\right)-C^i\right\|^4\right)$. with multiplier (β) = 0, SDG Mapping

$$\begin{aligned}&\min_{\alpha}\ln\left(\left(\left(mapping\left(C^i\right)-C^i\right)^t\left(mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right)\right)^2\right)\\&-\beta\ln\left(\alpha\left\|mapping\left(C^i\right)-C^i\right\|^2\right)\ln\left((1-\alpha)\left\|mapping\left(C^i\right)-C^i\right\|^2\right)\end{aligned}$$. with multiplier (β) = 0, SDG Mapping

$$\min_{\alpha} \ln\left(\left(\left(mapping\left(C^i\right)-C^i\right)^t\left(mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right)\right)^2\right) - \beta \ln\left(\alpha\left\|mapping\left(C^i\right)-C^i\right\|^2\right)\ln\left((1-\alpha)\left\|mapping\left(C^i\right)-C^i\right\|^2\right)$$
. with multiplier (β) = 0.001, SDG Mapping

$$\min_{\alpha} \ln\left(\left(\left(mapping\left(C^i\right)-C^i\right)^t\left(mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right)\right)^2\right) - \beta \ln\left(\alpha\left\|mapping\left(C^i\right)-C^i\right\|^2\right)\ln\left((1-\alpha)\left\|mapping\left(C^i\right)-C^i\right\|^2\right)$$
. with multiplier (β) = 0.01, SDG Mapping

$$\min_{\alpha} \ln\left(\left(\left(mapping\left(C^i\right)-C^i\right)^t\left(mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right)\right)^2\right) - \beta \ln\left(\alpha\left\|mapping\left(C^i\right)-C^i\right\|^2\right)\ln\left((1-\alpha)\left\|mapping\left(C^i\right)-C^i\right\|^2\right)$$
. with multiplier (β) = 0.1, SDG Mapping

$$\min_{\alpha} \ln\left(\left(\left(mapping\left(C^i\right)-C^i\right)^t\left(mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right)\right)^2\right) - \beta \ln\left(\alpha\left\|mapping\left(C^i\right)-C^i\right\|^2\right)\ln\left((1-\alpha)\left\|mapping\left(C^i\right)-C^i\right\|^2\right)$$
. with multiplier (β) = 1, SDG Mapping

Test Case Name	Network	Guidance	Path Splits	Flows	Times	Set of Code	Location in CD
`2_2_POTENTIALS, Network3, smaller`	link 102 -200 at 900 v/h smaller	100%	Aggregate	x10	x10	`CODE_2_2_POTENTIALS`	<CD Root>`/CODE_2_2_POTENTIALS`

$\min_{\alpha}\left\|mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right\|^2 - \beta\alpha(1-\alpha)\left\|mapping\left(C^i\right)-C^i\right\|^2$, with multiplier (β) = 0, DGS Mapping

$\min_{\alpha}\left\|mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right\|^2 - \beta\alpha(1-\alpha)\left\|mapping\left(C^i\right)-C^i\right\|^2$, with multiplier (β) = 0.001, DGS Mapping

$\min_{\alpha}\left\|mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right\|^2 - \beta\alpha(1-\alpha)\left\|mapping\left(C^i\right)-C^i\right\|^2$, with multiplier (β) = 0.01, DGS Mapping

$\min_{\alpha}\left\|mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right\|^2 - \beta\alpha(1-\alpha)\left\|mapping\left(C^i\right)-C^i\right\|^2$, with multiplier (β) = 0.1, DGS Mapping

$\min_{\alpha}\left\|mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right\|^2 - \beta\alpha(1-\alpha)\left\|mapping\left(C^i\right)-C^i\right\|^2$, with multiplier (β) = 1, DGS Mapping

$\min_{\alpha}\left\|mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right\|^2 - \beta\alpha^2\left\|mapping\left(C^i\right)-C^i\right\|^2$, with multiplier (β) = 0, DGS Mapping

$\min_{\alpha}\left\|mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right\|^2 - \beta\alpha^2\left\|mapping\left(C^i\right)-C^i\right\|^2$, with multiplier (β) = 0.001, DGS Mapping

$\min_{\alpha}\left\|mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right\|^2 - \beta\alpha^2\left\|mapping\left(C^i\right)-C^i\right\|^2$, with multiplier (β) = 0.01, DGS Mapping

$\min_{\alpha}\left\|mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right\|^2 - \beta\alpha^2\left\|mapping\left(C^i\right)-C^i\right\|^2$, with multiplier (β) = 0.1, DGS Mapping

$\min_{\alpha}\left\|mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right\|^2 - \beta\alpha^2\left\|mapping\left(C^i\right)-C^i\right\|^2$, with multiplier (β) = 1, DGS Mapping

$\min_{\alpha} \left\| mapping\left(C^i(\alpha)\right) - C^i(\alpha) \right\|^2 - \beta\alpha(1-\alpha)\left\| mapping\left(C^i\right) - C^i \right\|^2$, with multiplier (β) = 0, SDG Mapping

$\min_{\alpha} \left\| mapping\left(C^i(\alpha)\right) - C^i(\alpha) \right\|^2 - \beta\alpha(1-\alpha)\left\| mapping\left(C^i\right) - C^i \right\|^2$, with multiplier (β) = 0.001, SDG Mapping

$\min_{\alpha} \left\| mapping\left(C^i(\alpha)\right) - C^i(\alpha) \right\|^2 - \beta\alpha(1-\alpha)\left\| mapping\left(C^i\right) - C^i \right\|^2$, with multiplier (β) = 0.01, SDG Mapping

$\min_{\alpha} \left\| mapping\left(C^i(\alpha)\right) - C^i(\alpha) \right\|^2 - \beta\alpha(1-\alpha)\left\| mapping\left(C^i\right) - C^i \right\|^2$, with multiplier (β) = 0.1, SDG Mapping

$\min_{\alpha} \left\| mapping\left(C^i(\alpha)\right) - C^i(\alpha) \right\|^2 - \beta\alpha(1-\alpha)\left\| mapping\left(C^i\right) - C^i \right\|^2$, with multiplier (β) = 1, SDG Mapping

$\min_{\alpha} \left\| mapping\left(C^i(\alpha)\right) - C^i(\alpha) \right\|^2 - \beta\alpha^2\left\| mapping\left(C^i\right) - C^i \right\|^2$, with multiplier (β) = 0, SDG Mapping

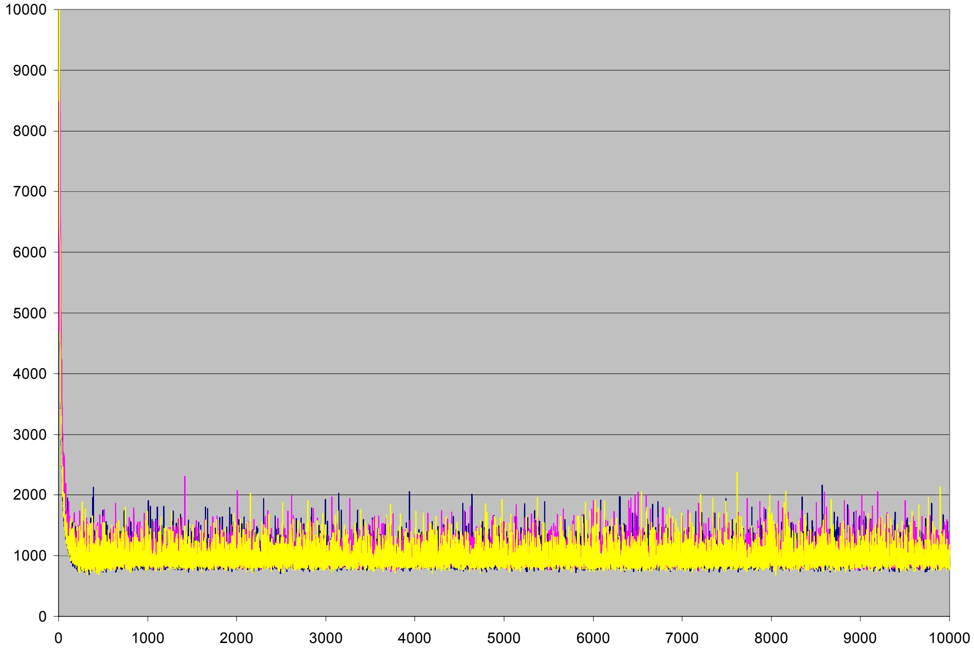
$\min_{\alpha} \left\| mapping\left(C^i(\alpha)\right) - C^i(\alpha) \right\|^2 - \beta\alpha^2\left\| mapping\left(C^i\right) - C^i \right\|^2$, with multiplier (β) = 0.001, SDG Mapping

$\min_{\alpha} \left\| mapping\left(C^i(\alpha)\right) - C^i(\alpha) \right\|^2 - \beta\alpha^2\left\| mapping\left(C^i\right) - C^i \right\|^2$, with multiplier (β) = 0.01, SDG Mapping

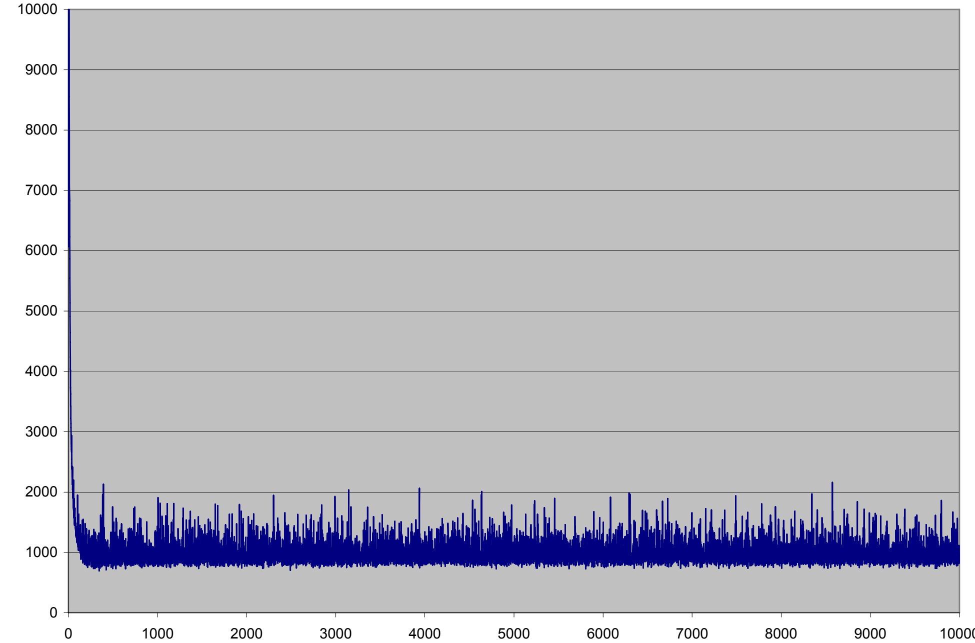
$\min_{\alpha} \left\| mapping\left(C^i(\alpha)\right) - C^i(\alpha) \right\|^2 - \beta\alpha^2\left\| mapping\left(C^i\right) - C^i \right\|^2$, with multiplier (β) = 0.1, SDG Mapping

$\min_{\alpha} \left\| mapping\left(C^i(\alpha)\right) - C^i(\alpha) \right\|^2 - \beta\alpha^2\left\| mapping\left(C^i\right) - C^i \right\|^2$, with multiplier (β) = 1, SDG Mapping

Test Case 2_0_10000_ITERATIONS, SDG Mapping

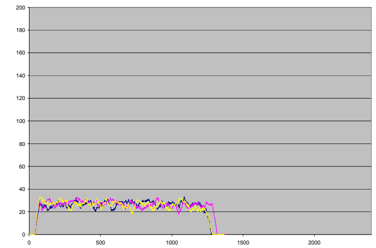


link time inconsistency norm

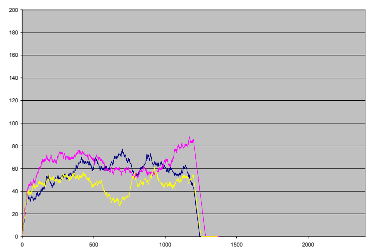


link time inconsistency norm

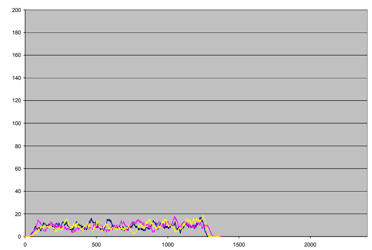
Link 4



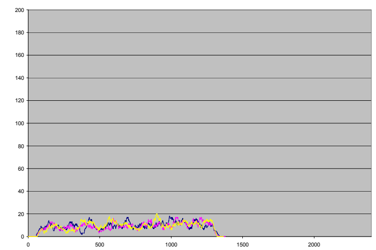
Link 1



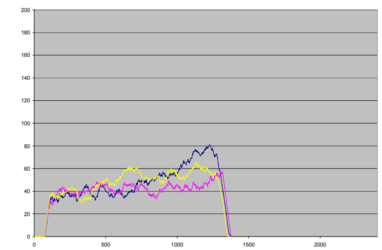
Link 5



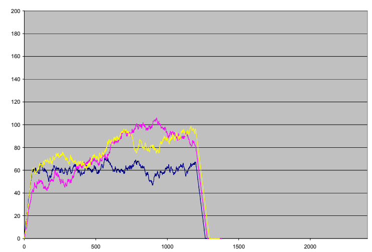
Link 8



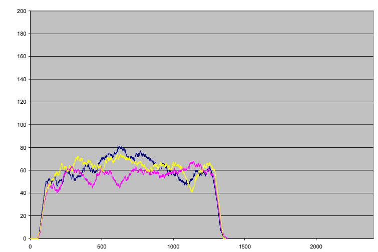
Link 6



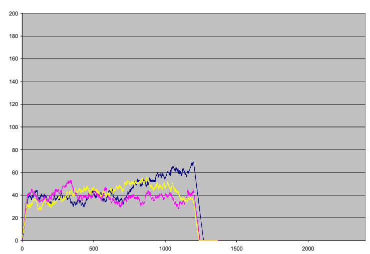
Link 2



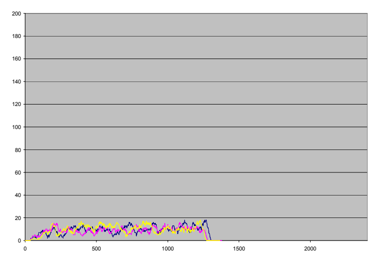
Link 9



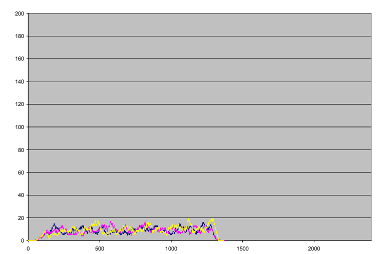
Link 3



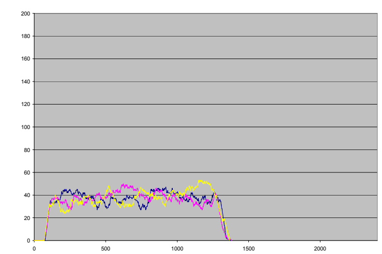
Link 11



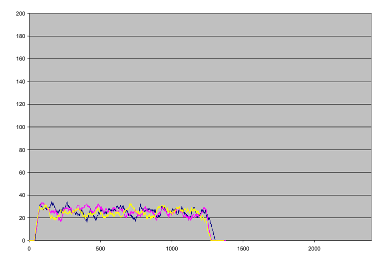
Link 10



Link 13

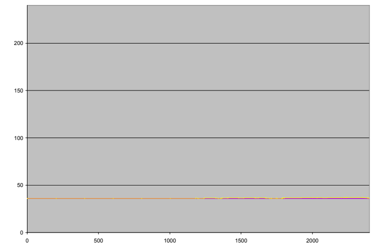


Link 12

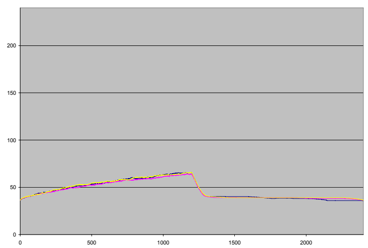


Link Volume Trajectories

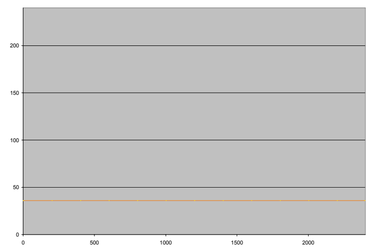
Link 4



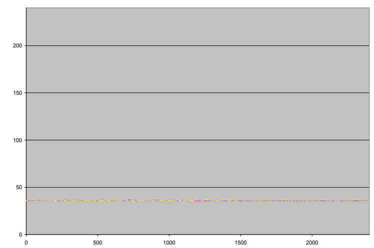
Link 1



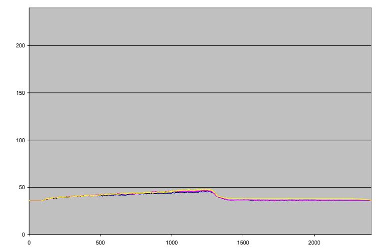
Link 5



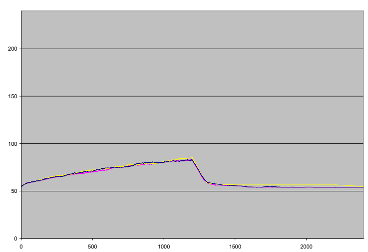
Link 8



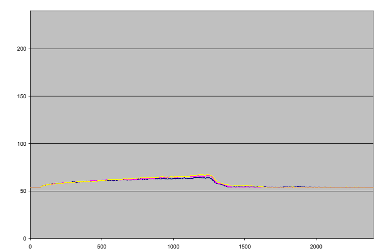
Link 6



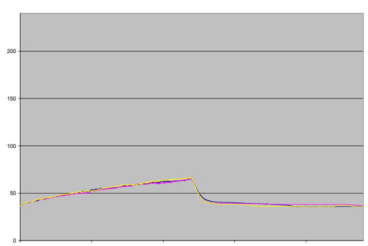
Link 2



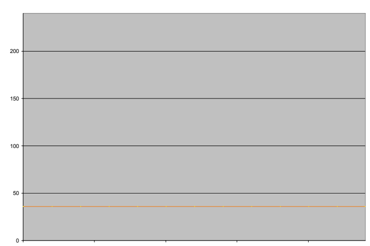
Link 9



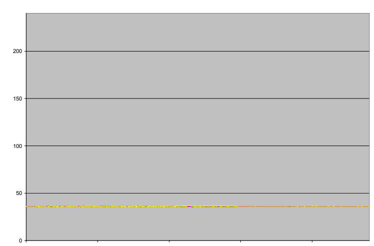
Link 3



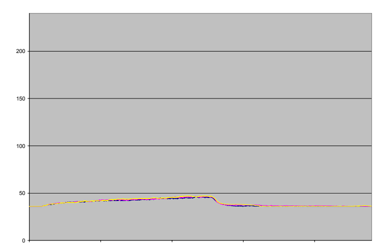
Link 11



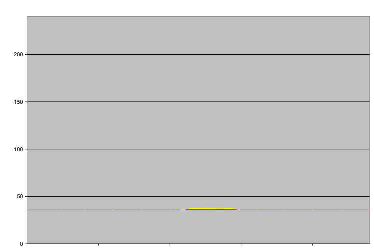
Link 10



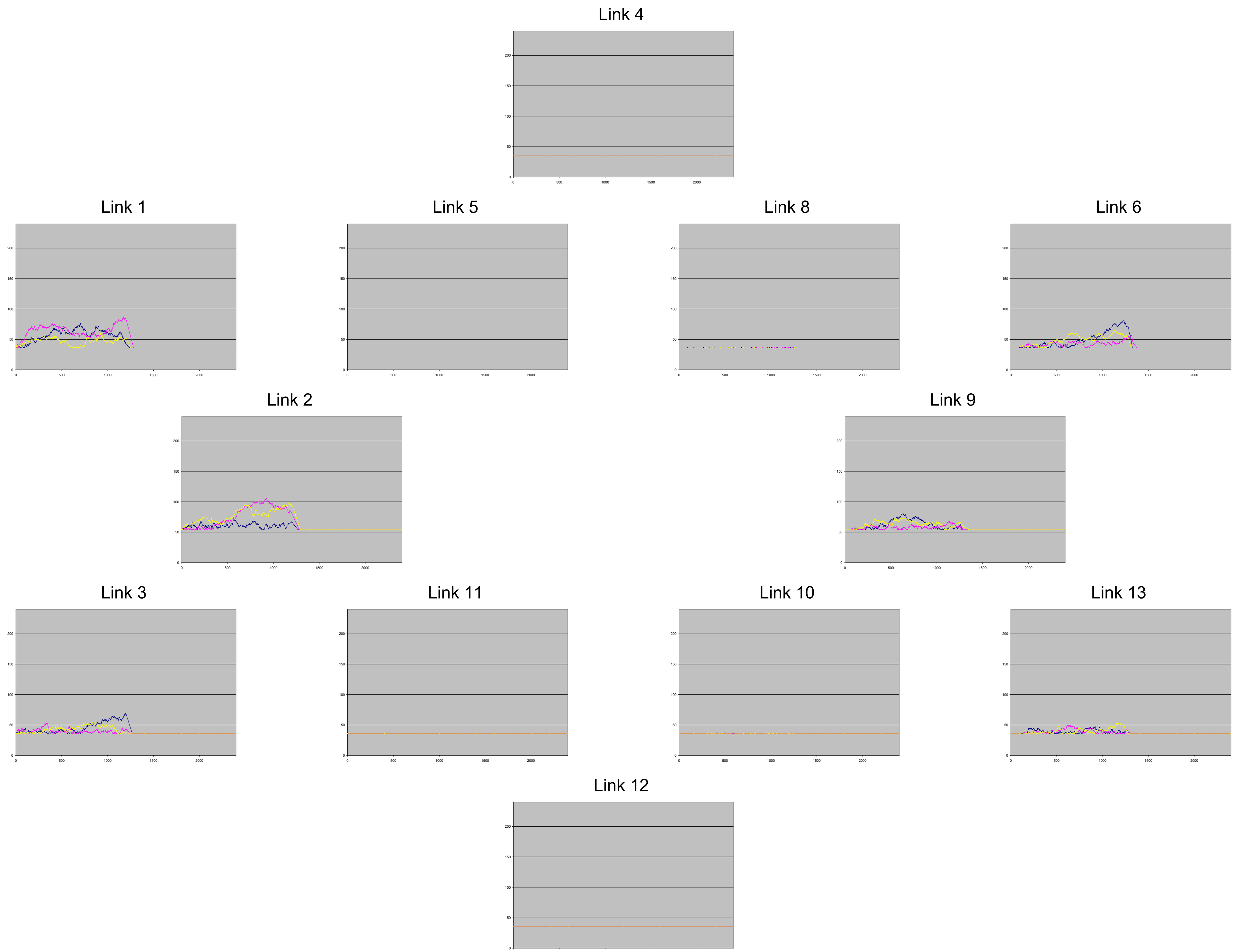
Link 13



Link 12

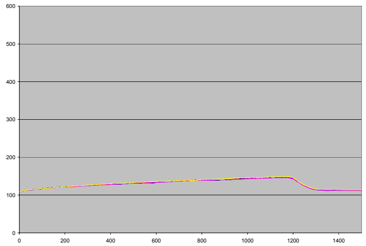


Input Link Traversal Time Trajectories

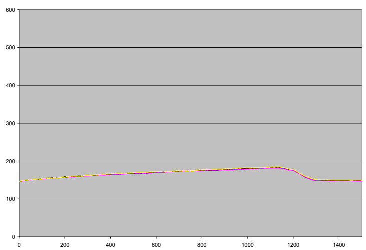


Output Link Traversal Time Trajectories

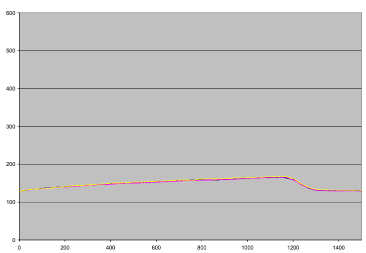
Path 0; links 1-4-6



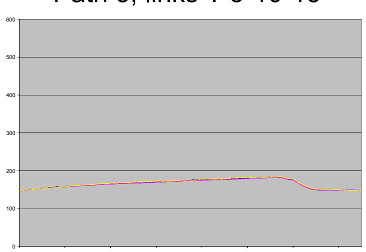
Path 1; links 1-5-8-6



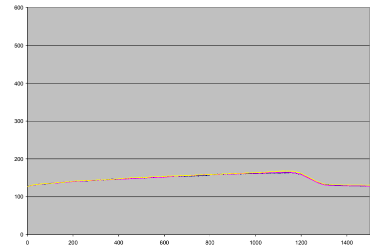
Path 2; links 1-5-9



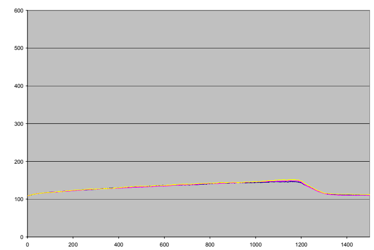
Path 3; links 1-5-10-13



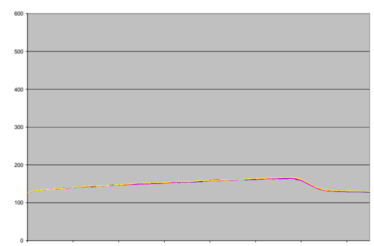
Path 5; links 2-8-6



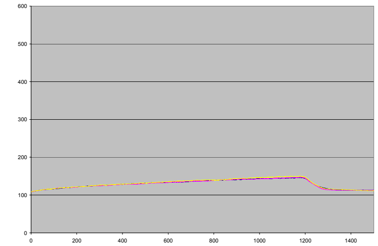
Path 4; links 2-9



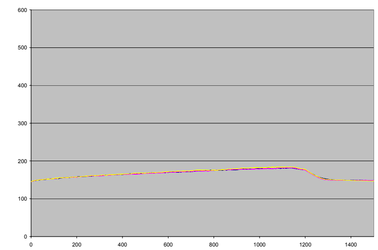
Path 6; links 2-10-13



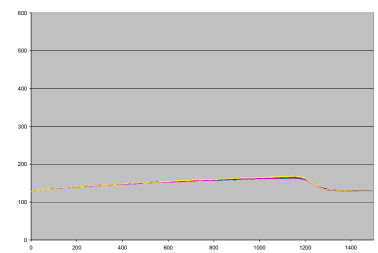
Path 7; links 3-12-13



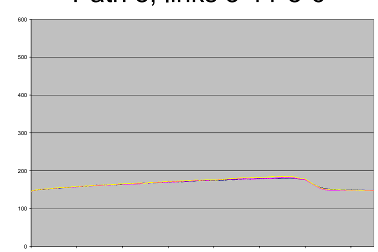
Path 10; links 3-11-10-13



Path 9; links 3-11-9

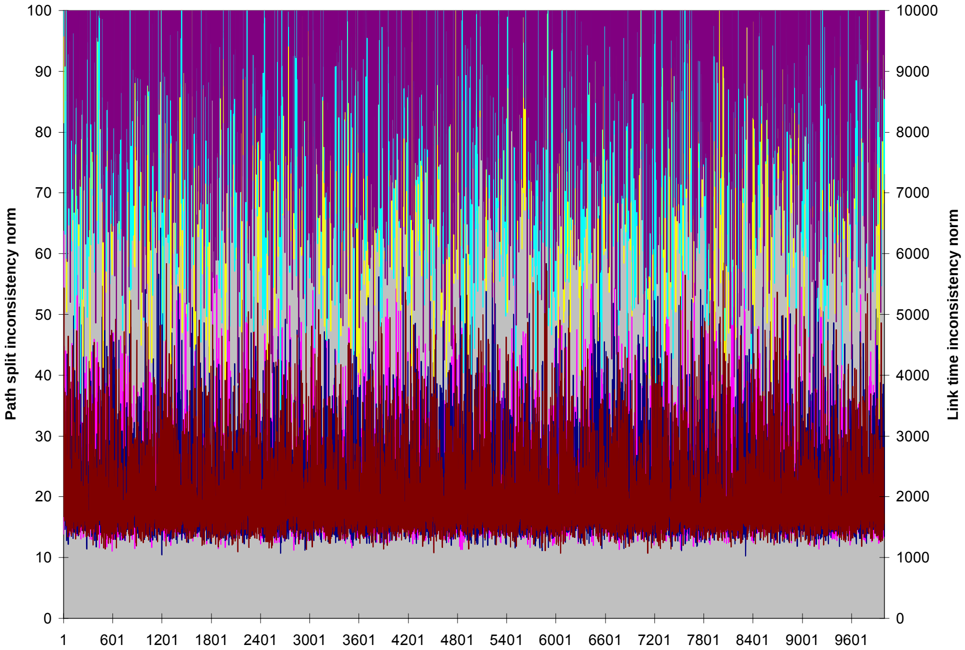


Path 8; links 3-11-8-6

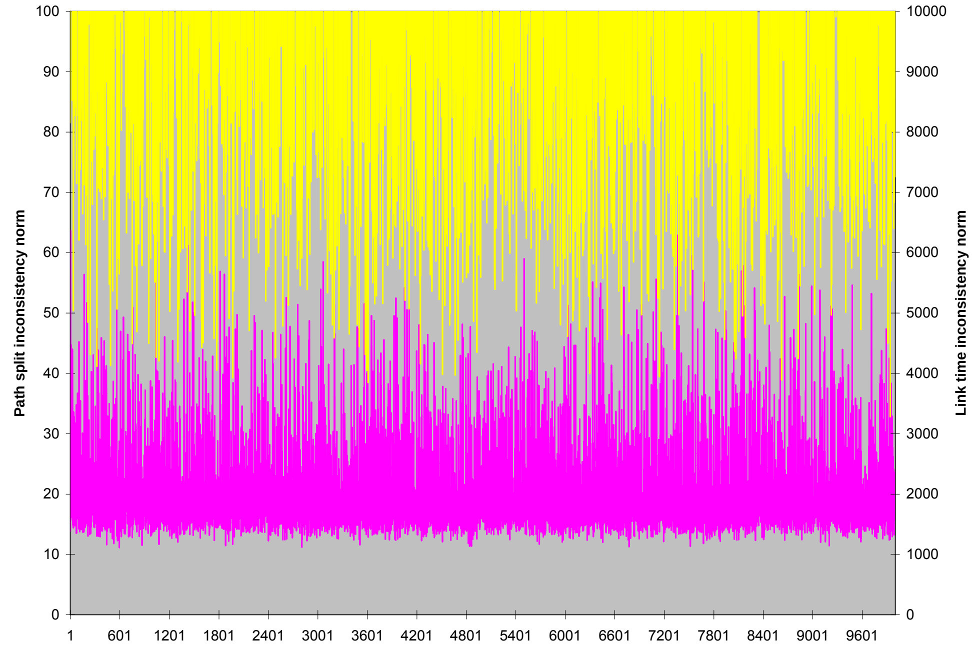


Path Time Trajectories

Test Case 2_0_10000_ITERATIONS, DGS Mapping

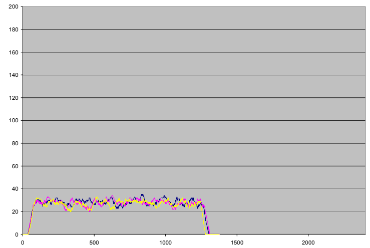


path split inconsistency norm and link time inconsistency norm

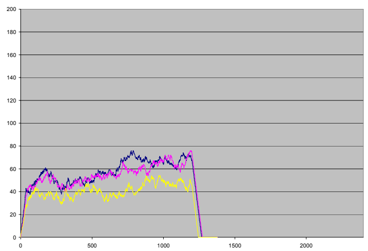


path split inconsistency norm and link time inconsistency norm (Replication1)

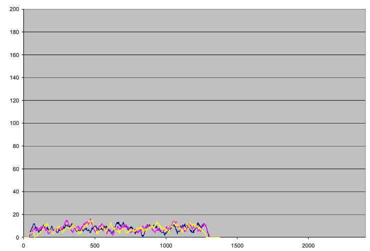
Link 4



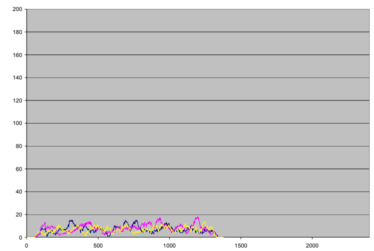
Link 1



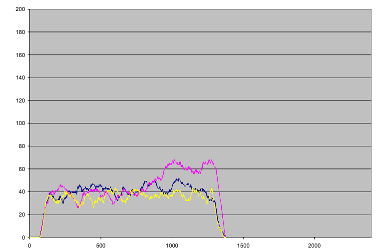
Link 5



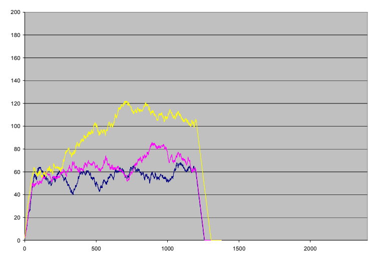
Link 8



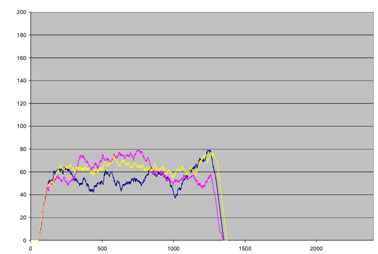
Link 6



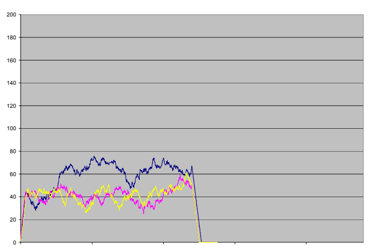
Link 2



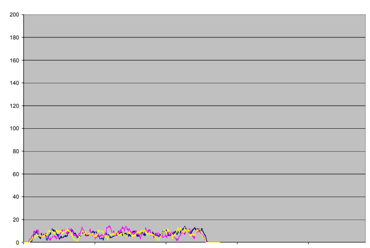
Link 9



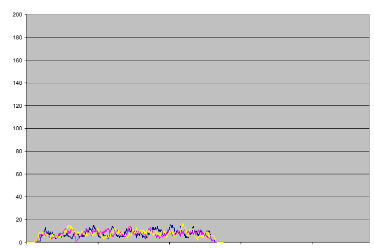
Link 3



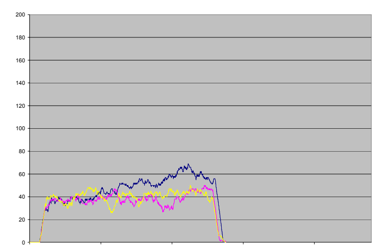
Link 11



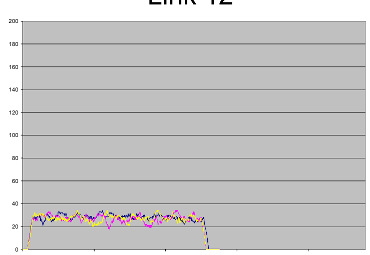
Link 10



Link 13

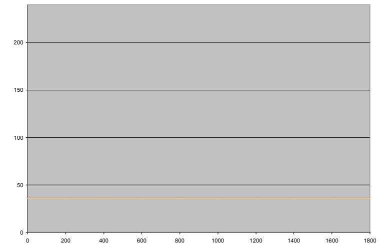


Link 12

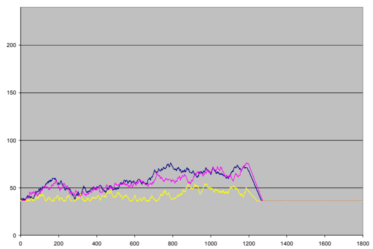


Link Volume Trajectories

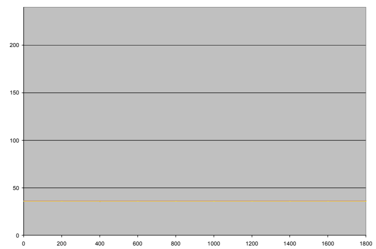
Link 4



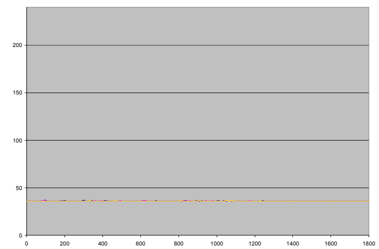
Link 1



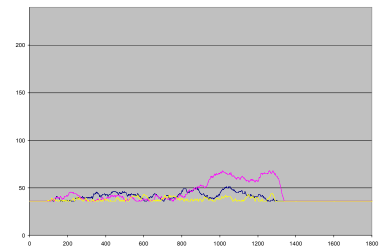
Link 5



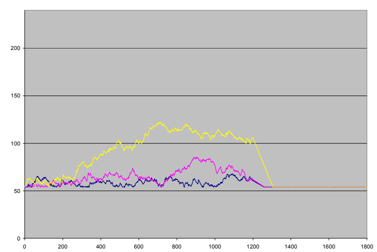
Link 8



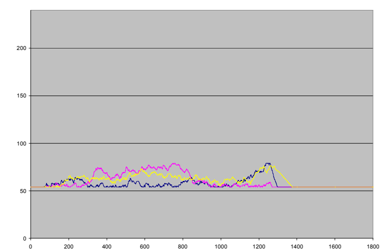
Link 6



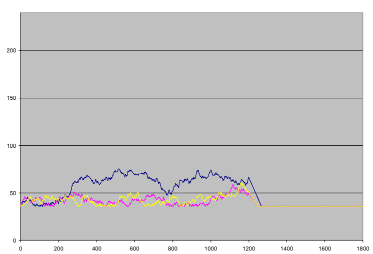
Link 2



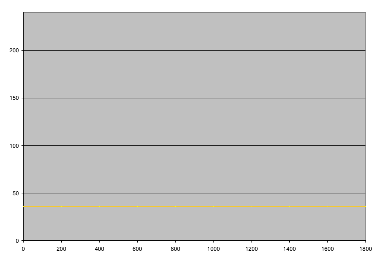
Link 9



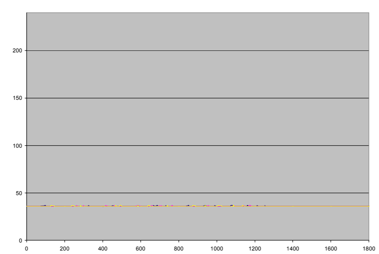
Link 3



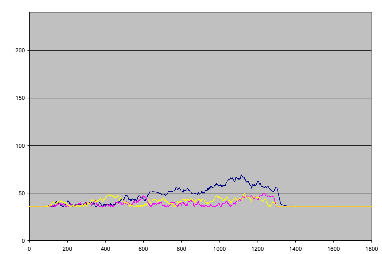
Link 11



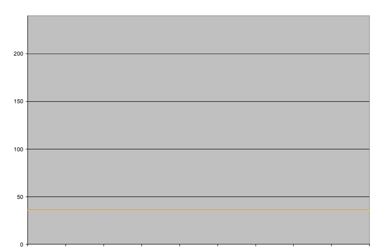
Link 10



Link 13

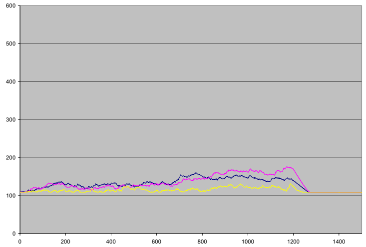


Link 12

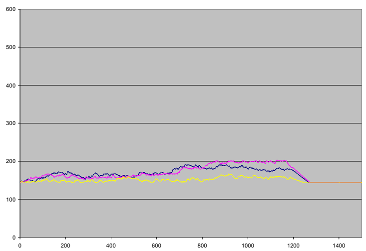


Link Time Trajectories

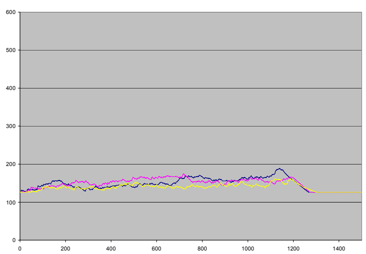
Path 0; links 1-4-6



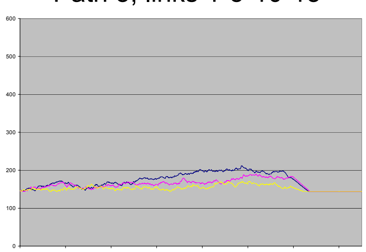
Path 1; links 1-5-8-6



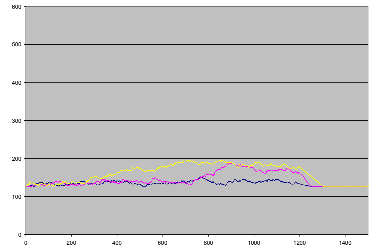
Path 2; links 1-5-9



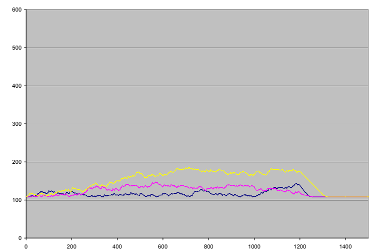
Path 3; links 1-5-10-13



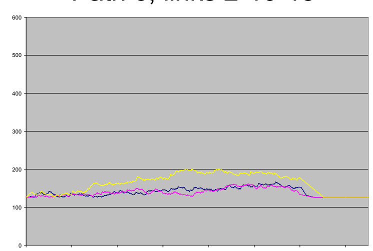
Path 5; links 2-8-6



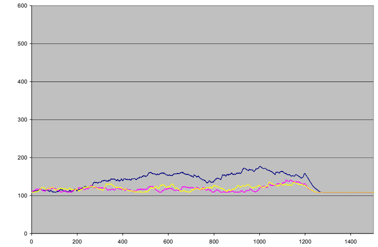
Path 4; links 2-9



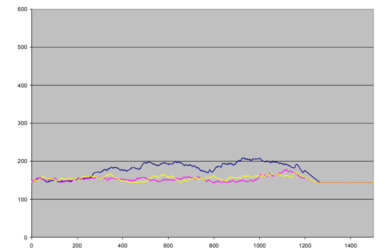
Path 6; links 2-10-13



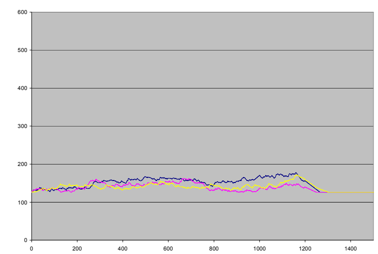
Path 7; links 3-12-13



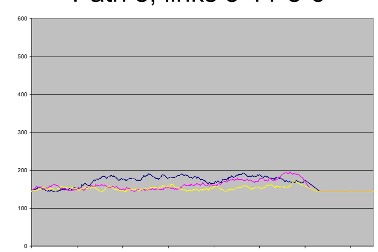
Path 10; links 3-11-10-13



Path 9; links 3-11-9

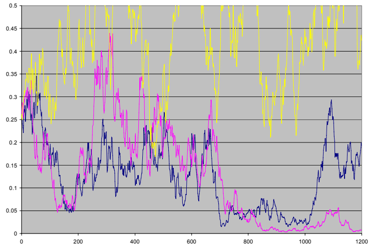


Path 8; links 3-11-8-6

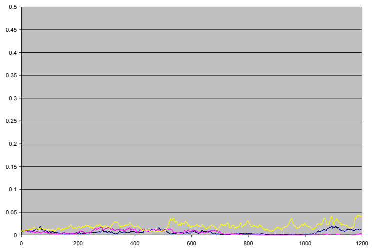


Path Time Trajectories

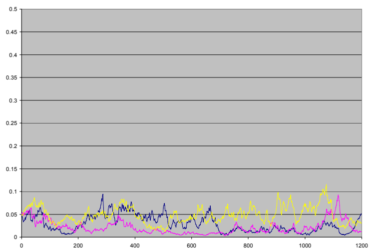
Path 0; links 1-4-6



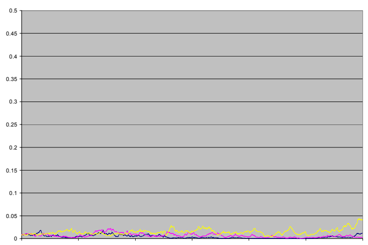
Path 1; links 1-5-8-6



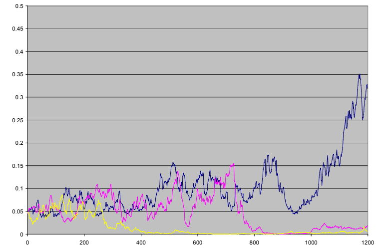
Path 2; links 1-5-9



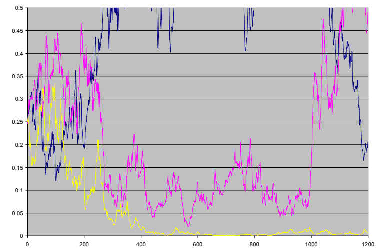
Path 3; links 1-5-10-13



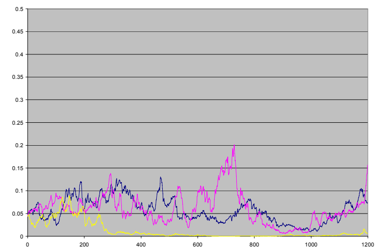
Path 5; links 2-8-6



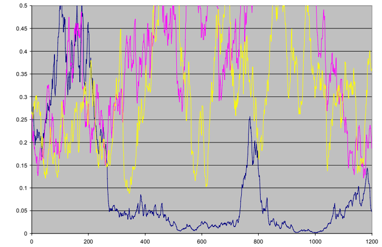
Path 4; links 2-9



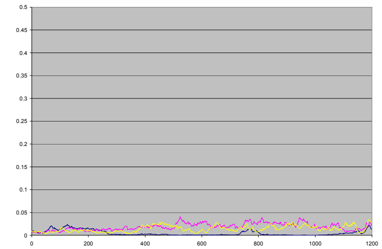
Path 6; links 2-10-13



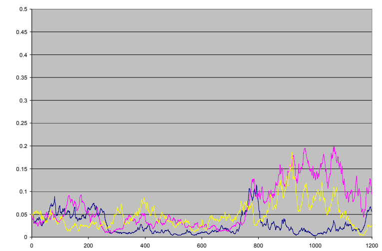
Path 7; links 3-12-13



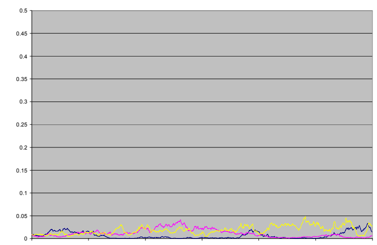
Path 10; links 3-11-10-13



Path 9; links 3-11-9

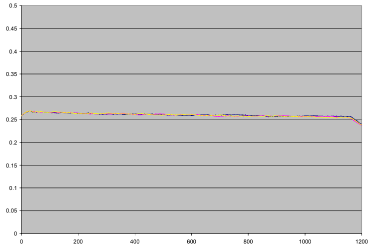


Path 8; links 3-11-8-6

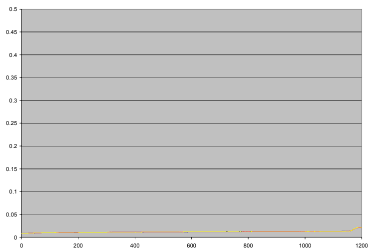


Output Pre-Trip Path Split Trajectories

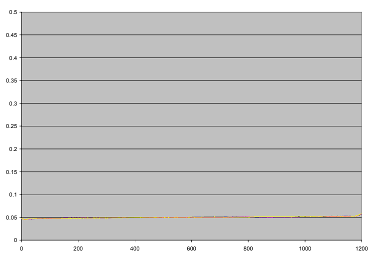
Path 0; links 1-4-6



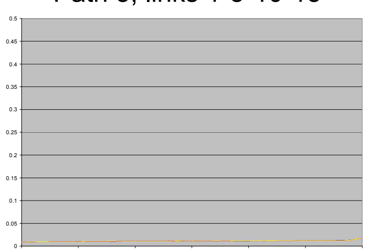
Path 1; links 1-5-8-6



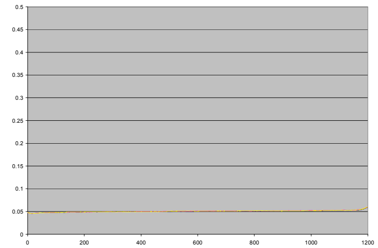
Path 2; links 1-5-9



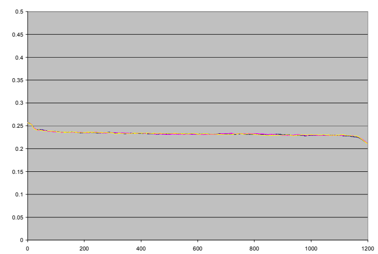
Path 3; links 1-5-10-13



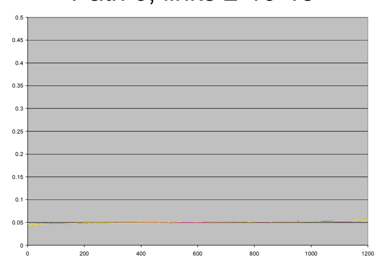
Path 5; links 2-8-6



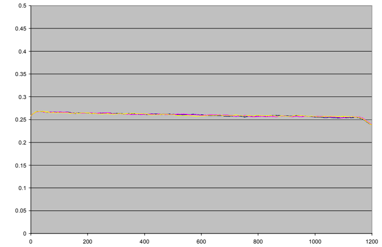
Path 4; links 2-9



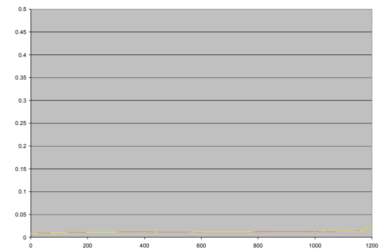
Path 6; links 2-10-13



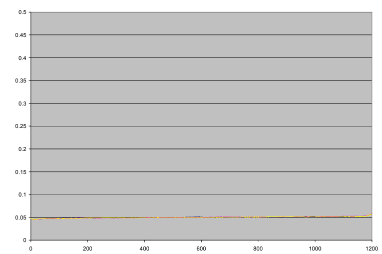
Path 7; links 3-12-13



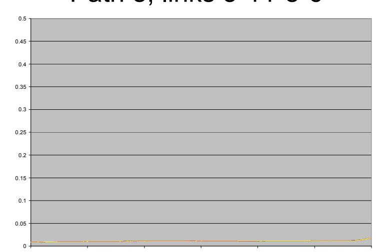
Path 10; links 3-11-10-13



Path 9; links 3-11-9

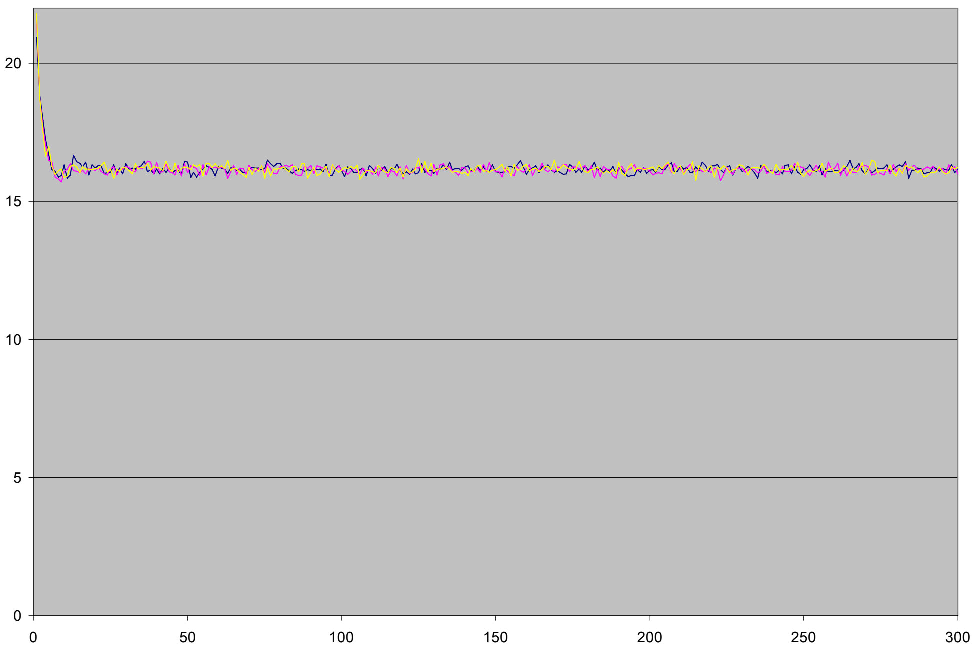


Path 8; links 3-11-8-6

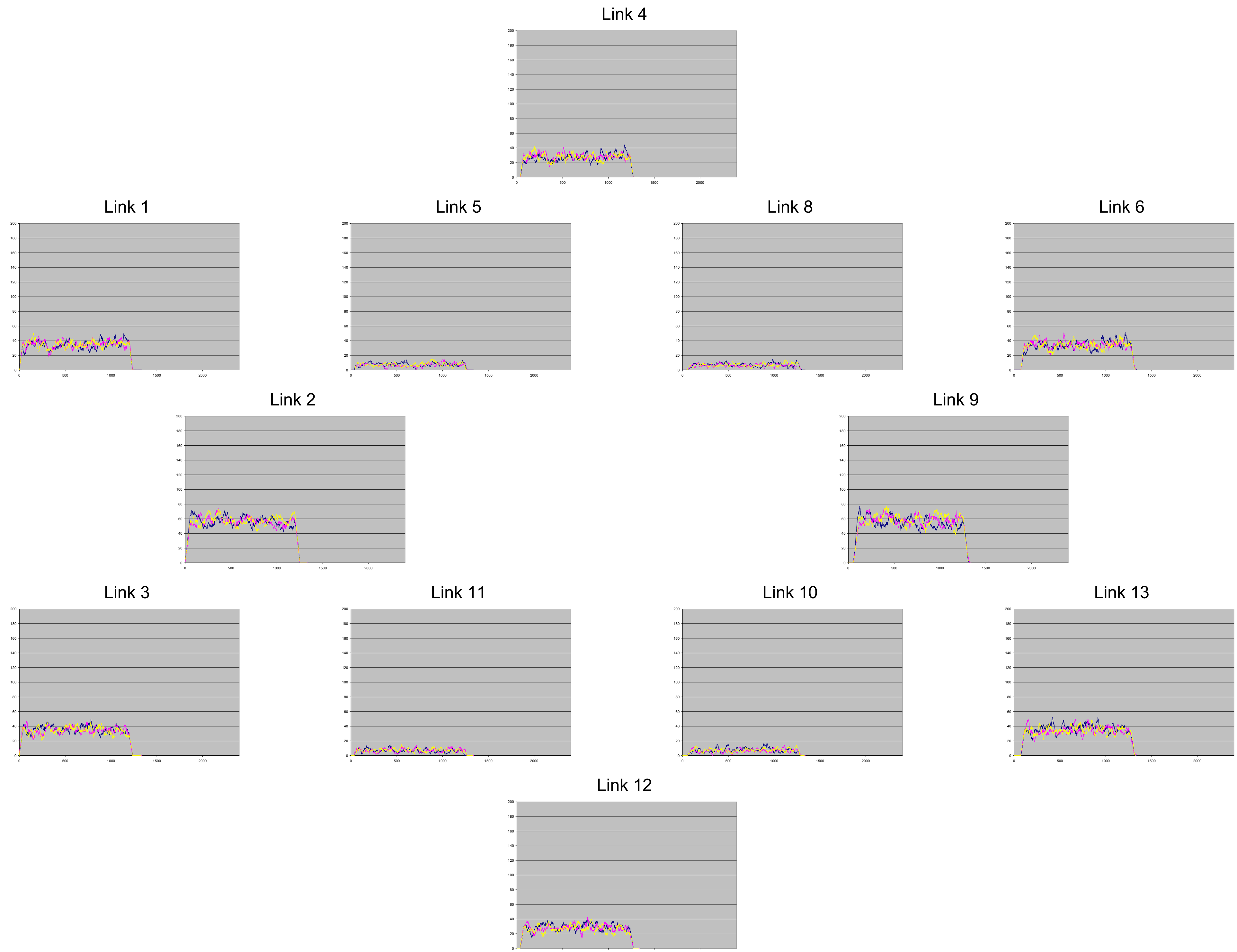


Input Pre-Trip Path Split Trajectories

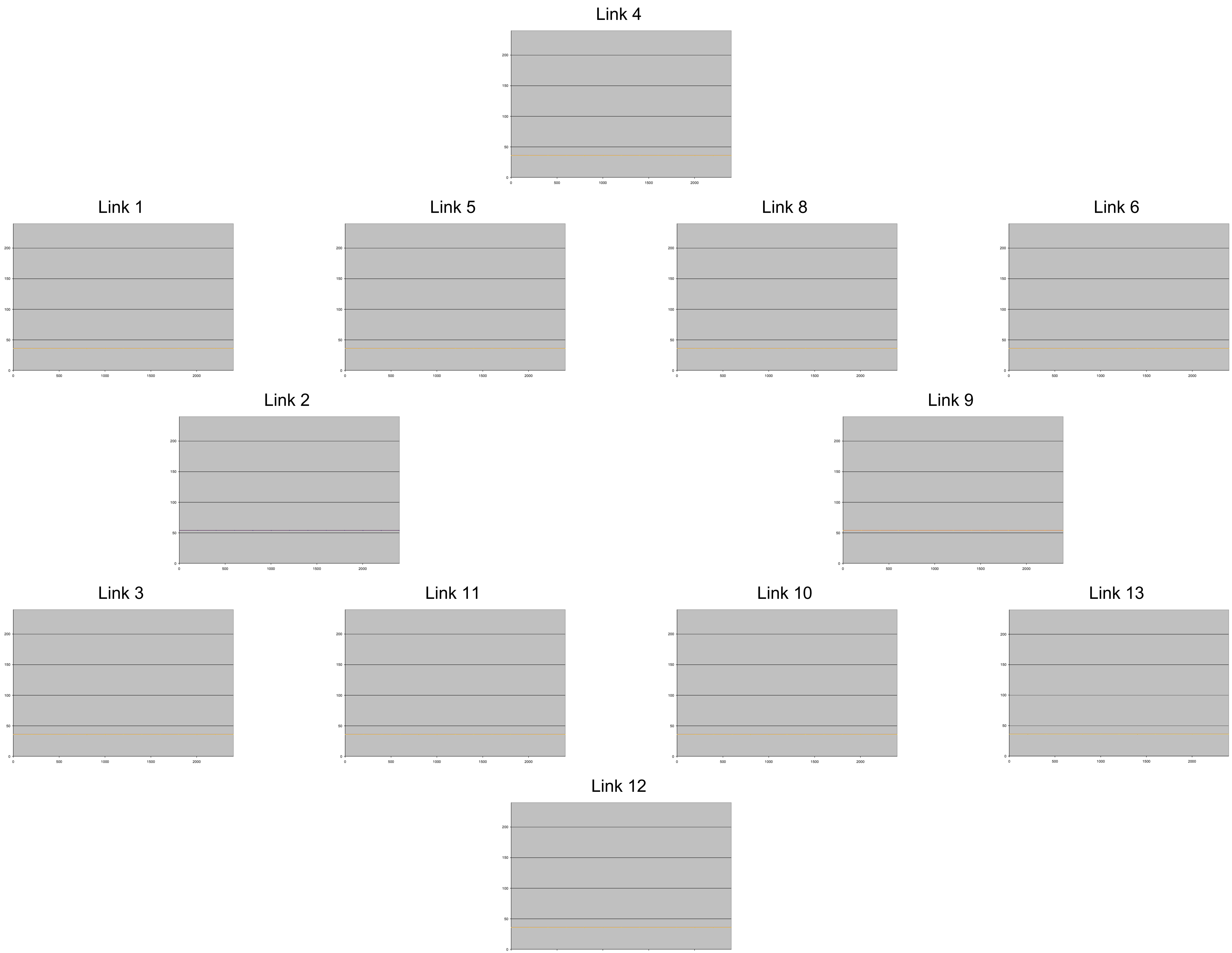
Test Case 2_0_RELAXED_EXIT_CAPACITY, SDG Mapping



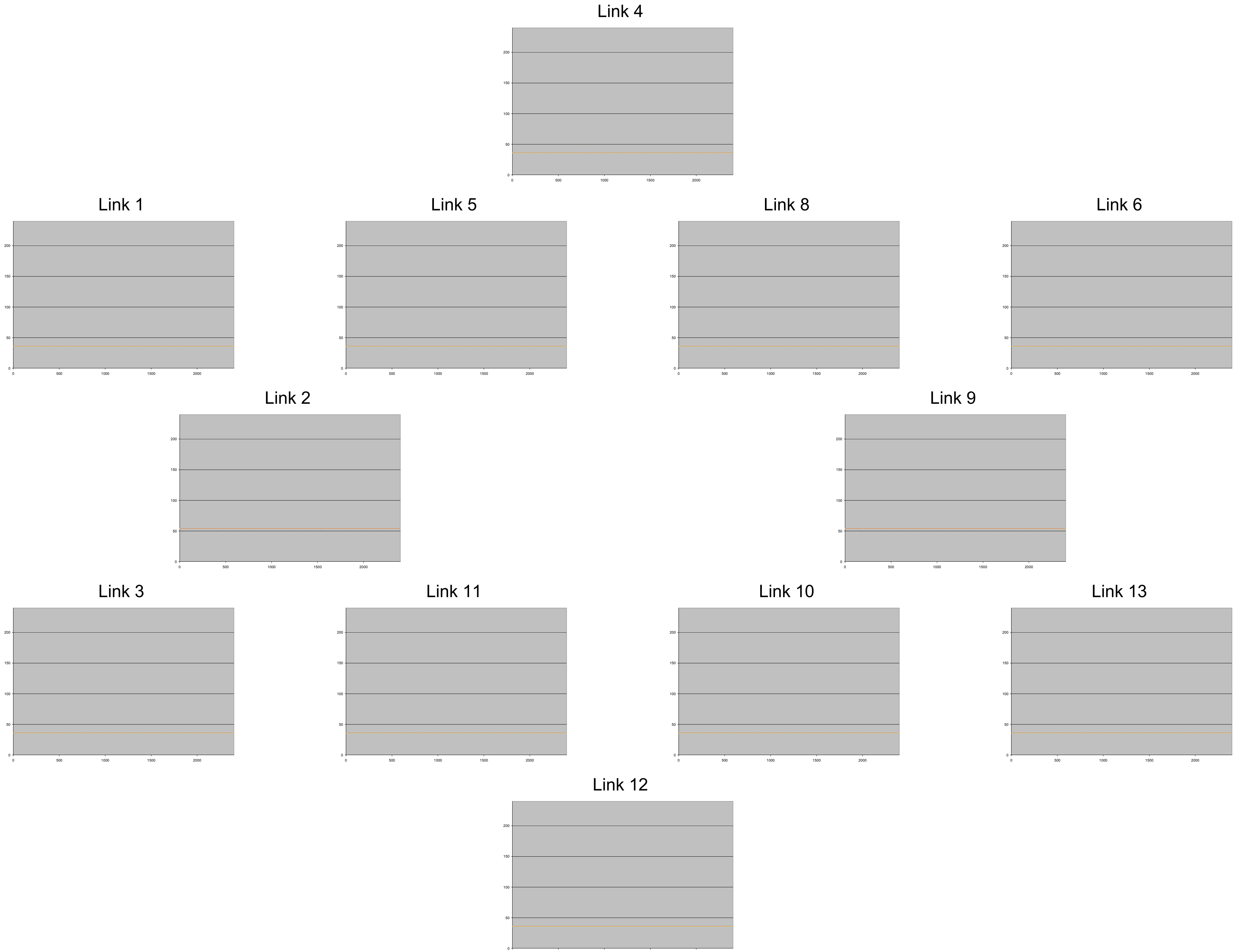
link time inconsistency norm



Link Volume Trajectories

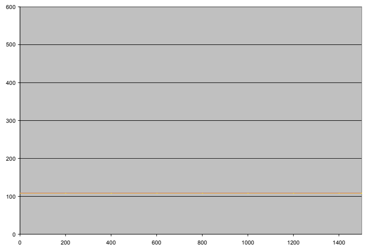


Input Link Traversal Time Trajectories

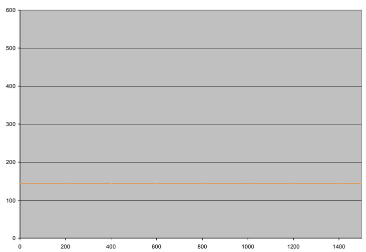


Output Link Traversal Time Trajectories

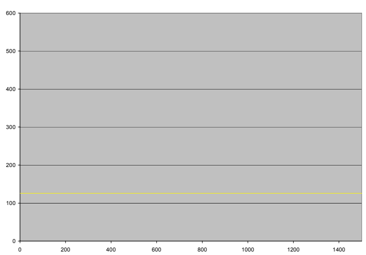
Path 0; links 1-4-6



Path 1; links 1-5-8-6

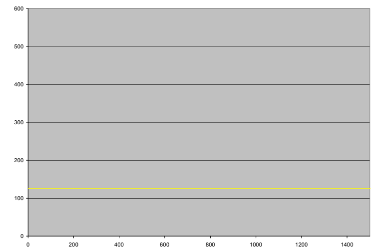


Path 2; links 1-5-9

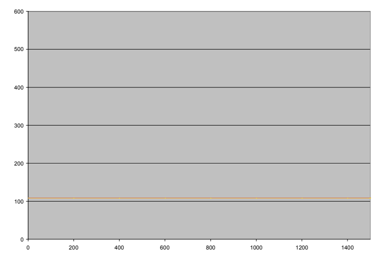


Path 3; links 1-5-10-13

Path 5; links 2-8-6

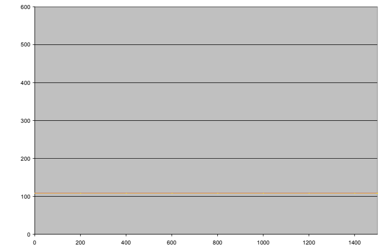


Path 4; links 2-9

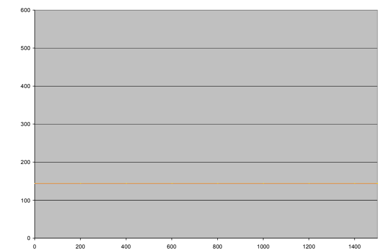


Path 6; links 2-10-13

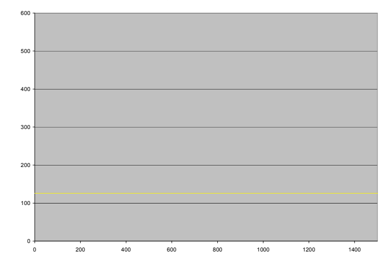
Path 7; links 3-12-13



Path 10; links 3-11-10-13



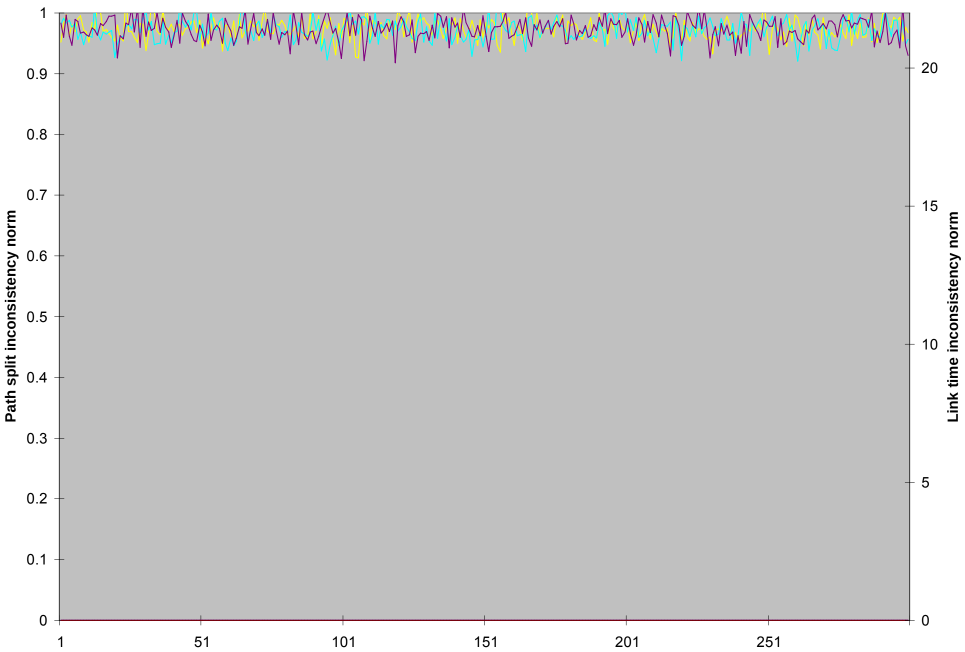
Path 9; links 3-11-9



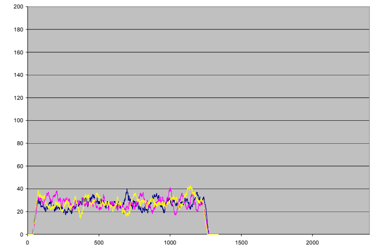
Path 8; links 3-11-8-6

Path Time Trajectories

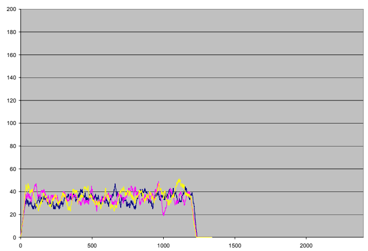
Test Case 2_0_RELAXED_EXIT_CAPACITY, DGS Mapping



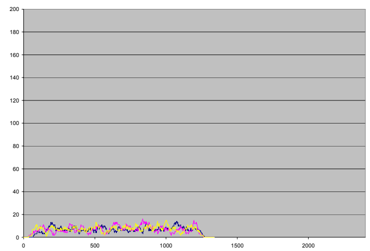
Link 4



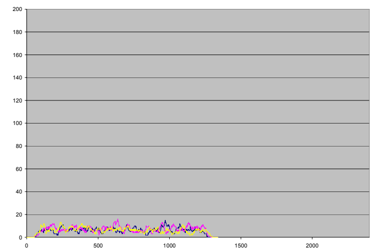
Link 1



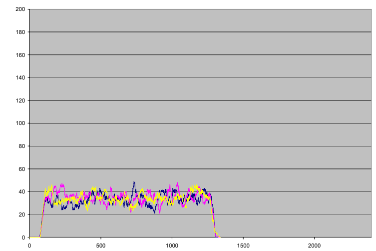
Link 5



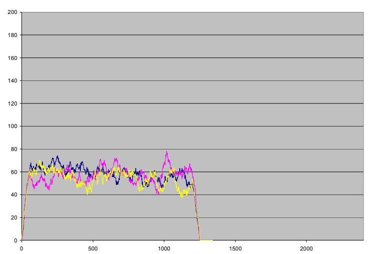
Link 8



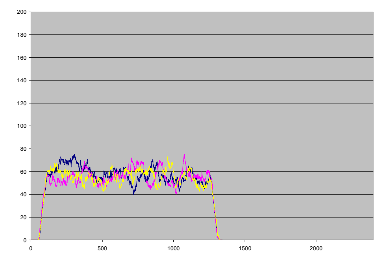
Link 6



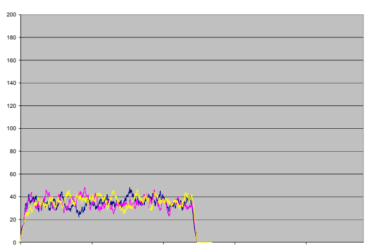
Link 2



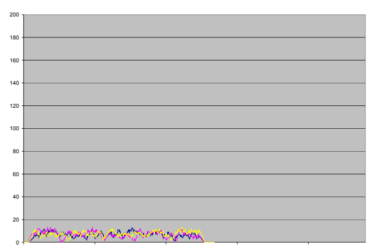
Link 9



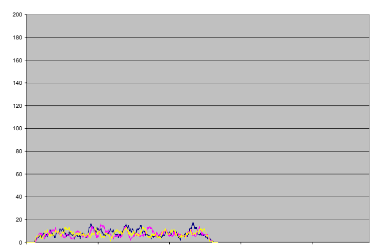
Link 3



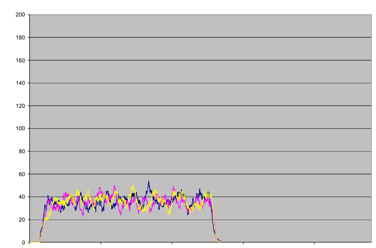
Link 11



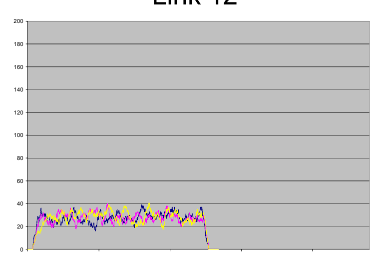
Link 10



Link 13

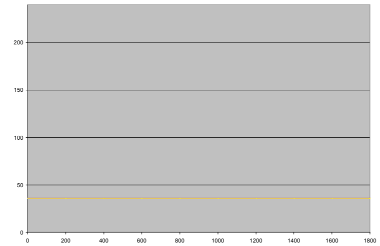


Link 12

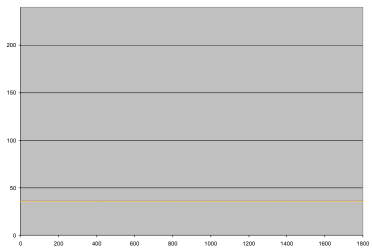


Link Volume Trajectories

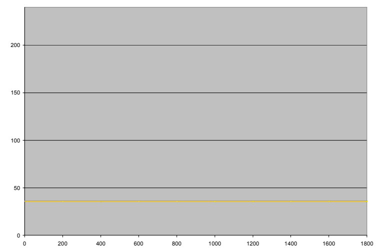
Link 4



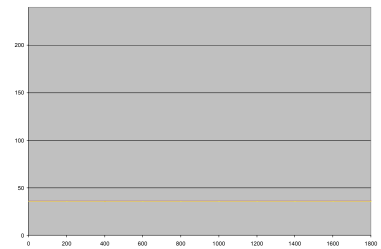
Link 1



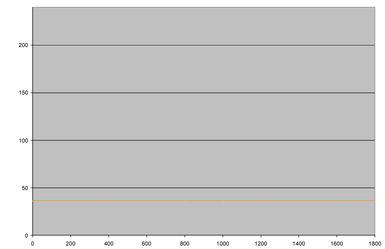
Link 5



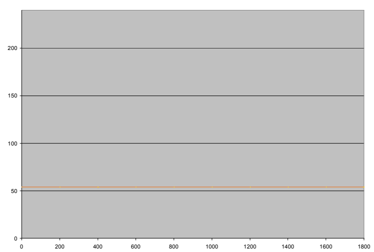
Link 8



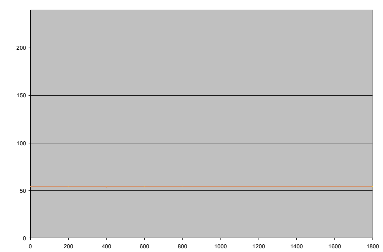
Link 6



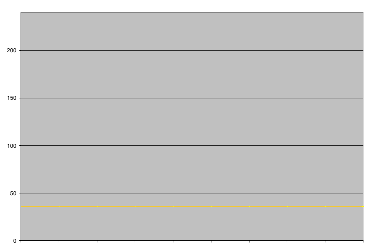
Link 2



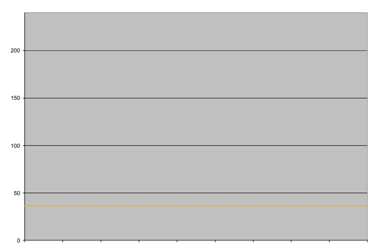
Link 9



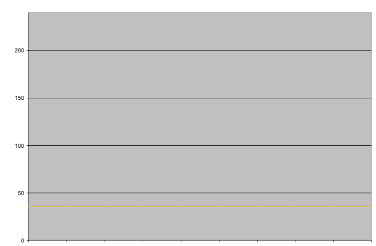
Link 3



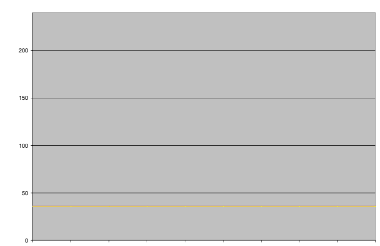
Link 11



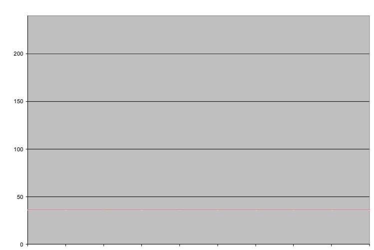
Link 10



Link 13

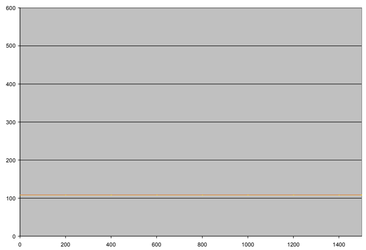


Link 12

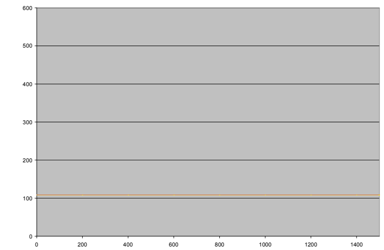


Link Time Trajectories

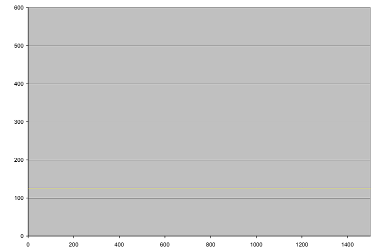
Path 0; links 1-4-6



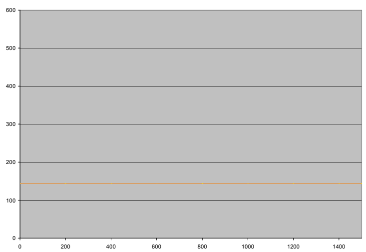
Path 7; links 3-12-13



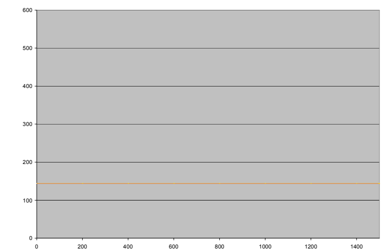
Path 5; links 2-8-6



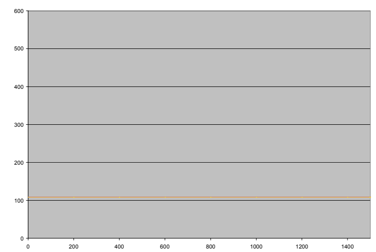
Path 1; links 1-5-8-6



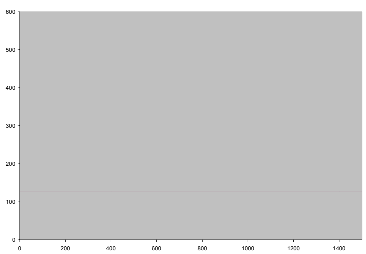
Path 10; links 3-11-10-13



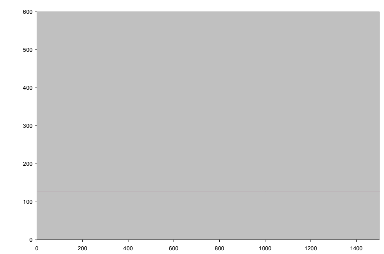
Path 4; links 2-9



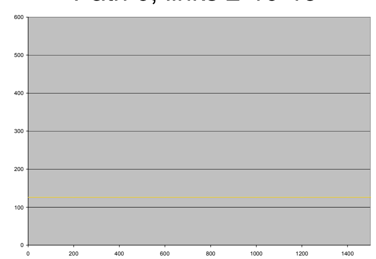
Path 2; links 1-5-9



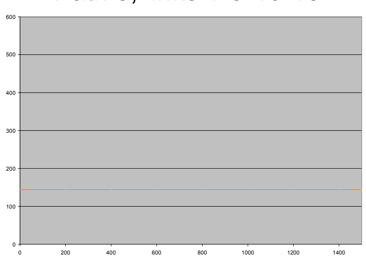
Path 9; links 3-11-9



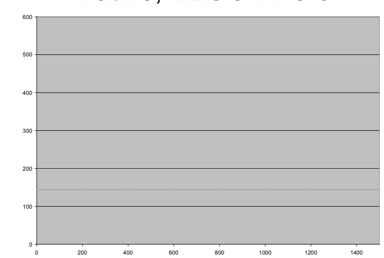
Path 6; links 2-10-13



Path 3; links 1-5-10-13

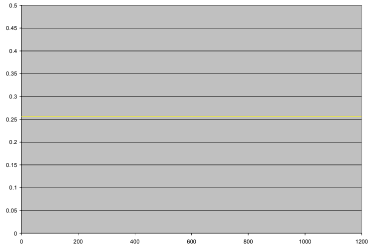


Path 8; links 3-11-8-6

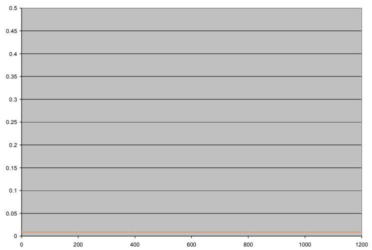


Path Time Trajectories

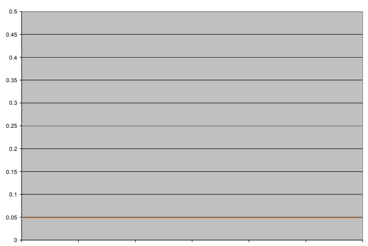
Path 0; links 1-4-6



Path 1; links 1-5-8-6

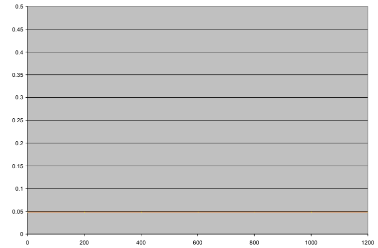


Path 2; links 1-5-9

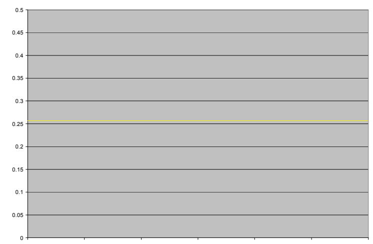


Path 3; links 1-5-10-13

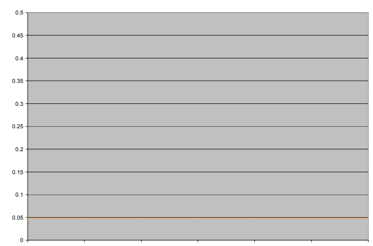
Path 5; links 2-8-6



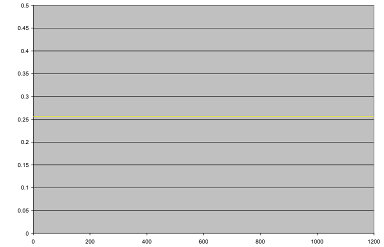
Path 4; links 2-9



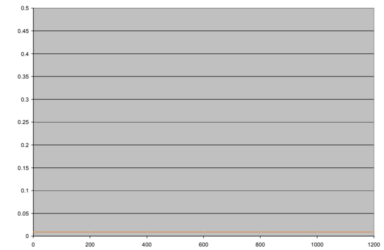
Path 6; links 2-10-13



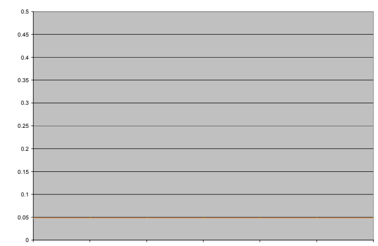
Path 7; links 3-12-13



Path 10; links 3-11-10-13



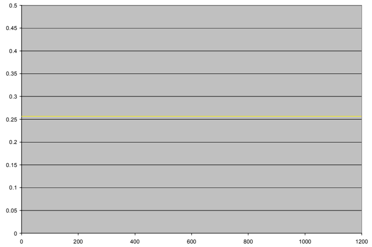
Path 9; links 3-11-9



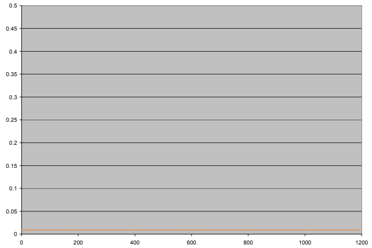
Path 8; links 3-11-8-6

Output Pre-Trip Path Split Trajectories

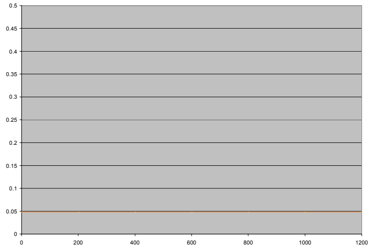
Path 0; links 1-4-6



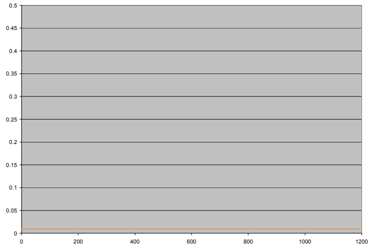
Path 1; links 1-5-8-6



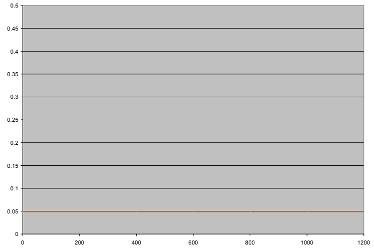
Path 2; links 1-5-9



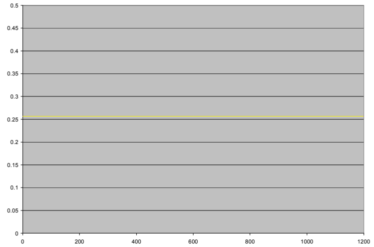
Path 3; links 1-5-10-13



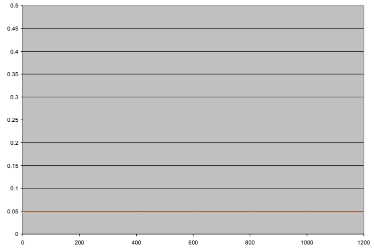
Path 5; links 2-8-6



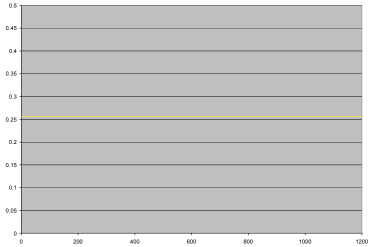
Path 4; links 2-9



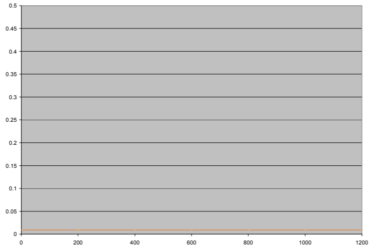
Path 6; links 2-10-13



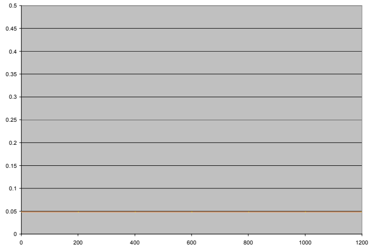
Path 7; links 3-12-13



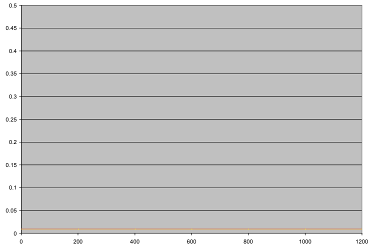
Path 10; links 3-11-10-13



Path 9; links 3-11-9

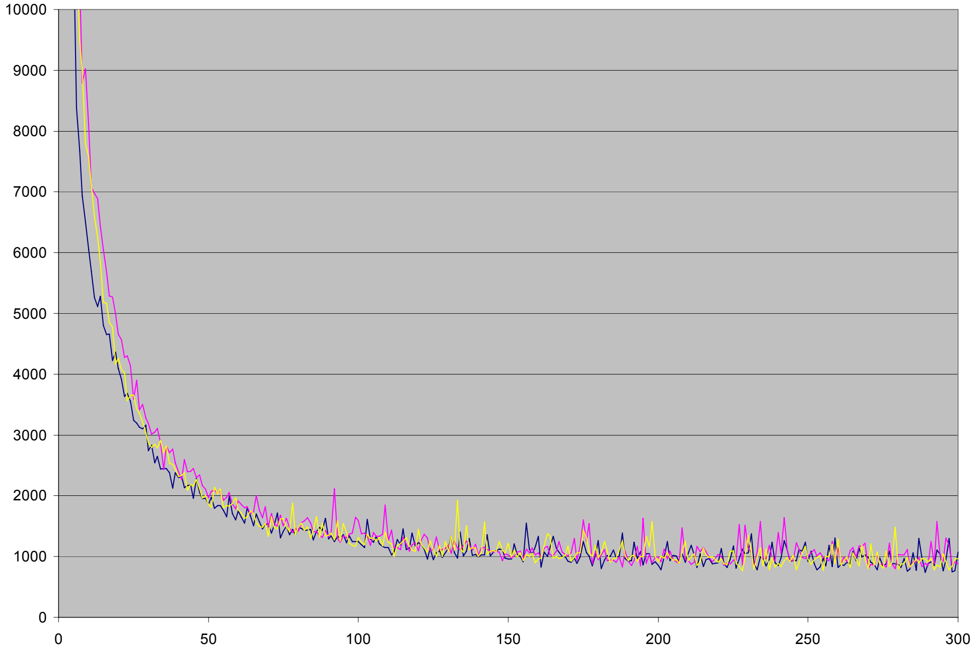


Path 8; links 3-11-8-6



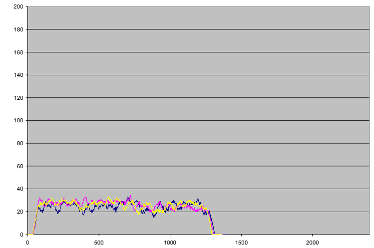
Input Pre-Trip Path Split Trajectories

Test Case 2_0_RELAXED_STORAGE_CAPACITY, SDG Mapping

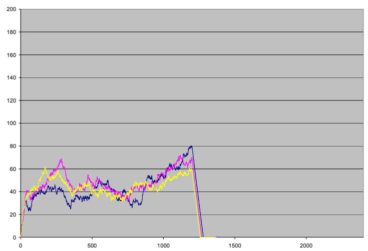


link time inconsistency norm

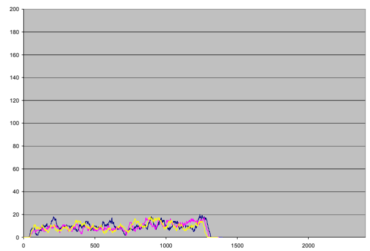
Link 4



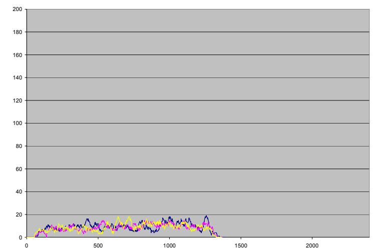
Link 1



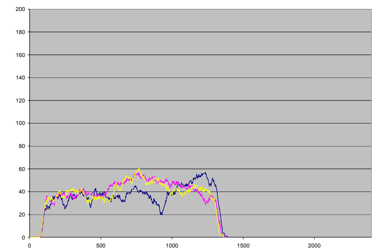
Link 5



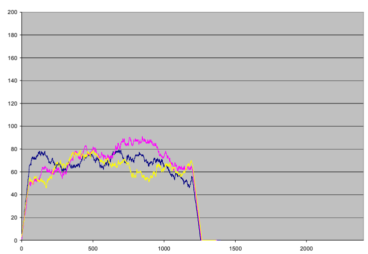
Link 8



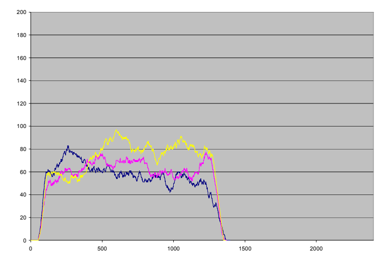
Link 6



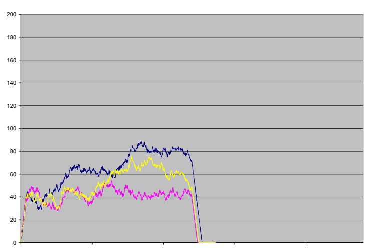
Link 2



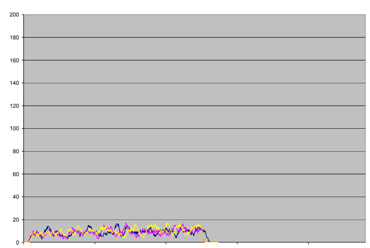
Link 9



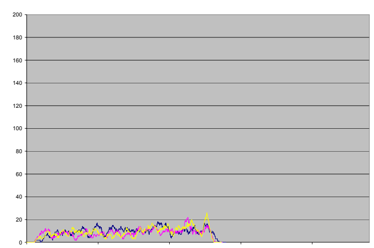
Link 3



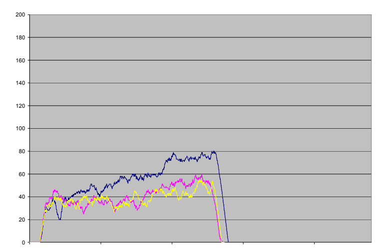
Link 11



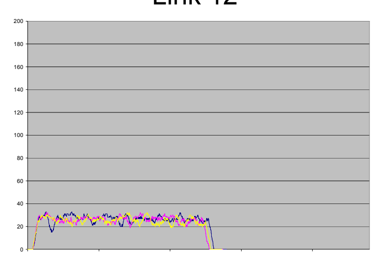
Link 10



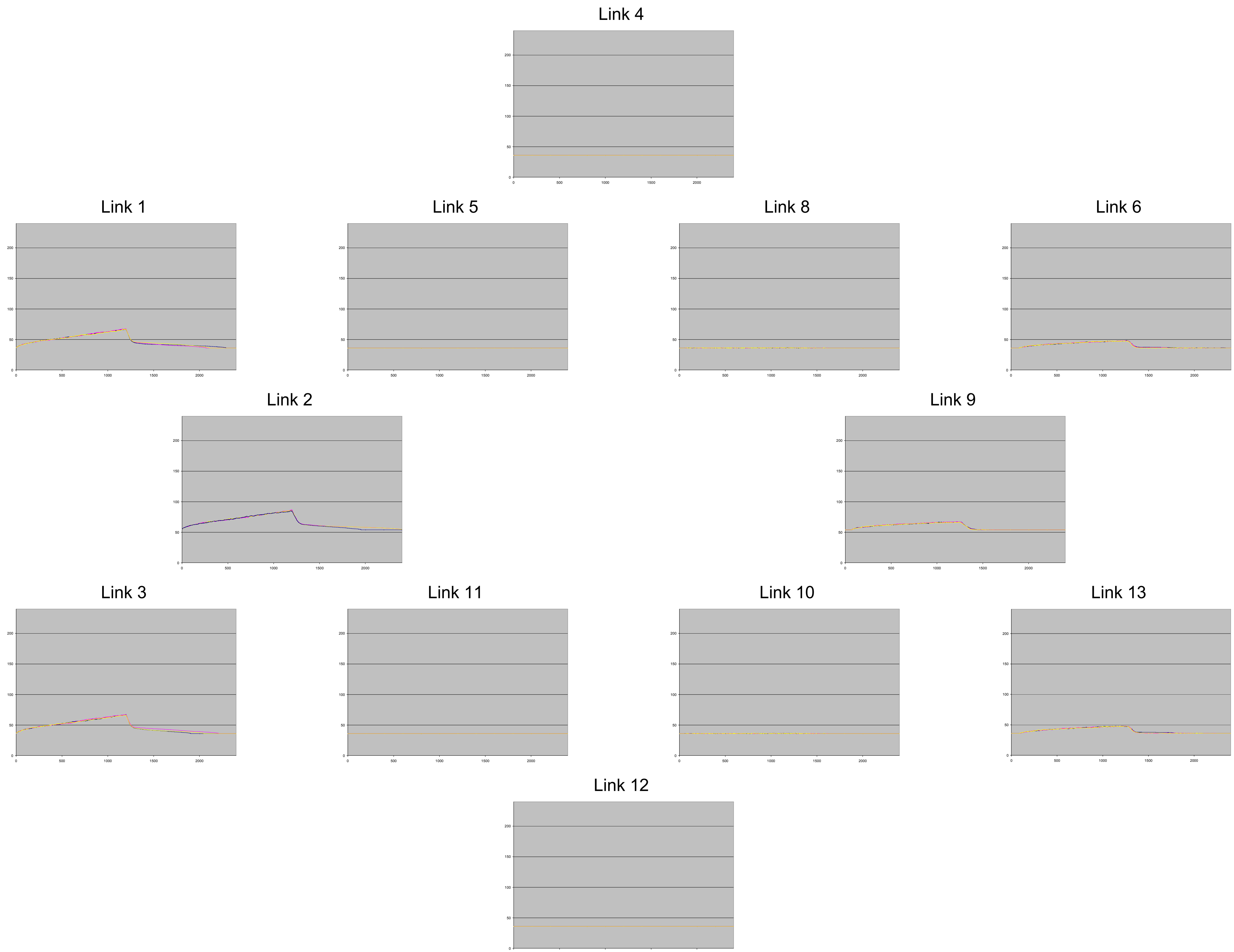
Link 13



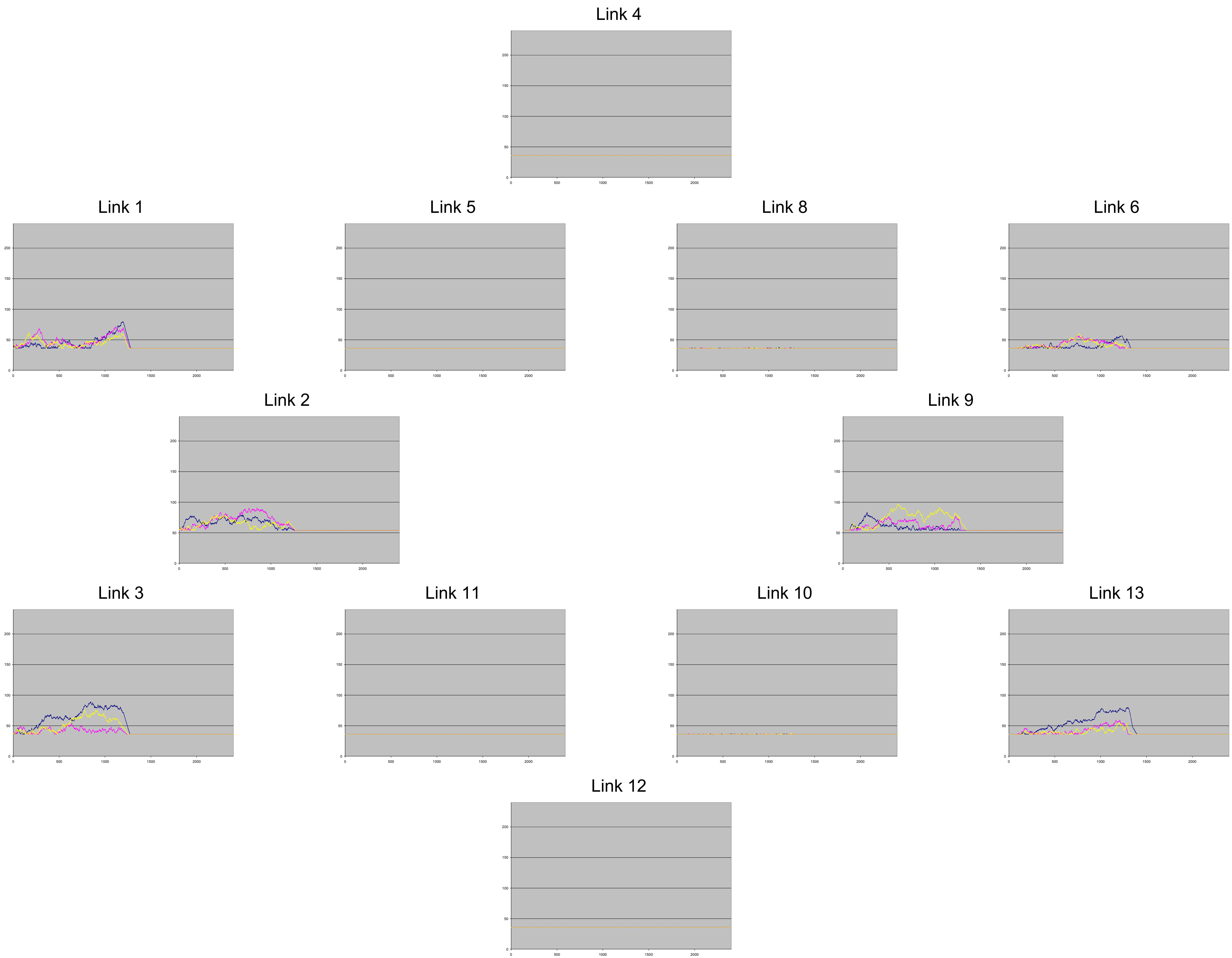
Link 12



Link Volume Trajectories

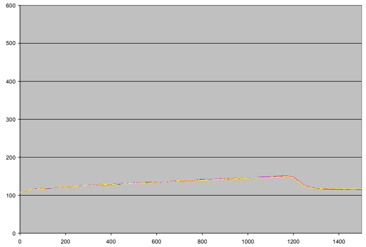


Input Link Traversal Time Trajectories

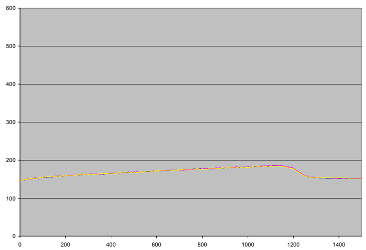


Output Link Traversal Time Trajectories

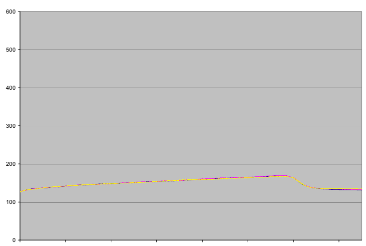
Path 0; links 1-4-6



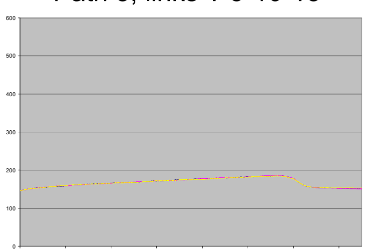
Path 1; links 1-5-8-6



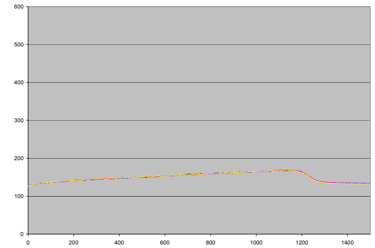
Path 2; links 1-5-9



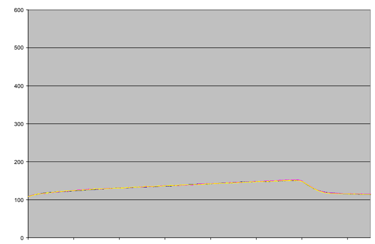
Path 3; links 1-5-10-13



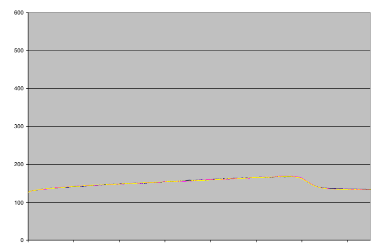
Path 5; links 2-8-6



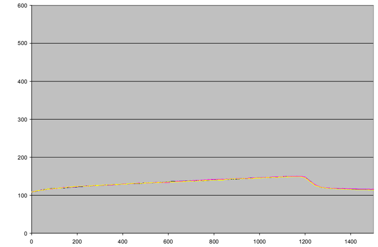
Path 4; links 2-9



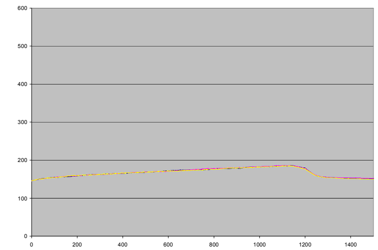
Path 6; links 2-10-13



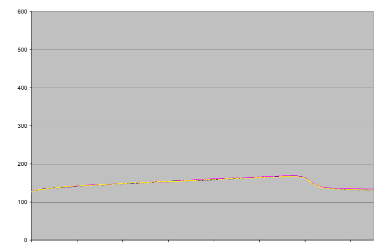
Path 7; links 3-12-13



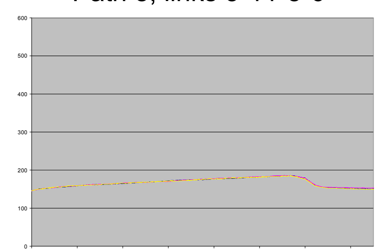
Path 10; links 3-11-10-13



Path 9; links 3-11-9

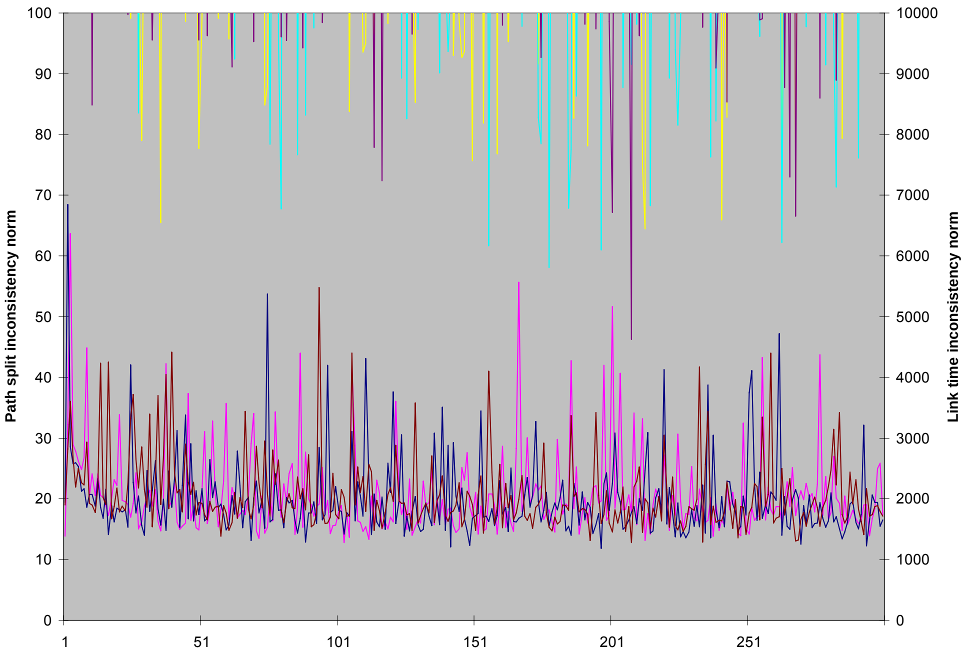


Path 8; links 3-11-8-6

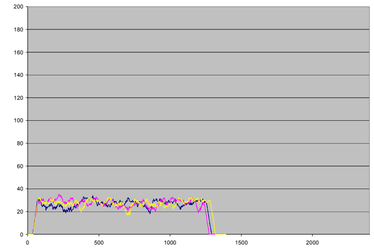


Path Time Trajectories

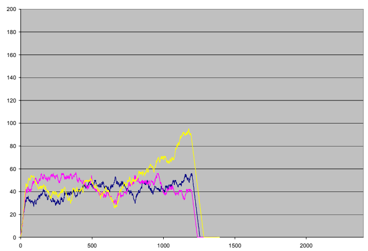
Test Case 2_0_RELAXED_STORAGE_CAPACITY, DGS Mapping



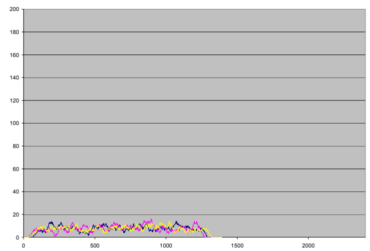
Link 4



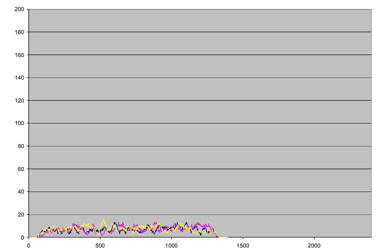
Link 1



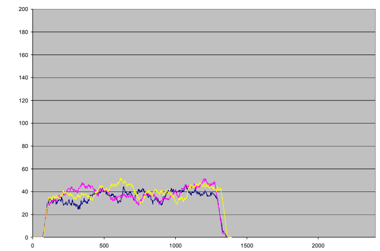
Link 5



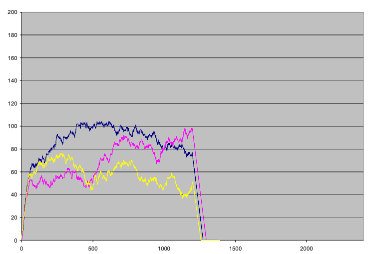
Link 8



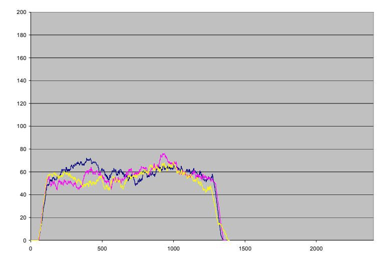
Link 6



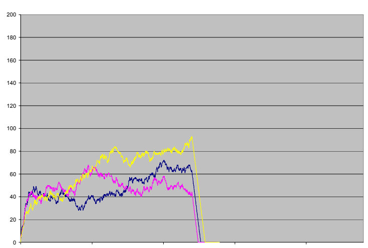
Link 2



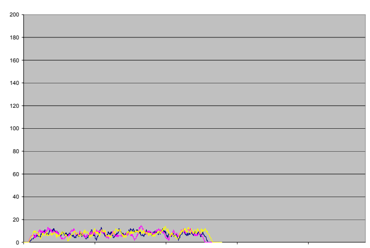
Link 9



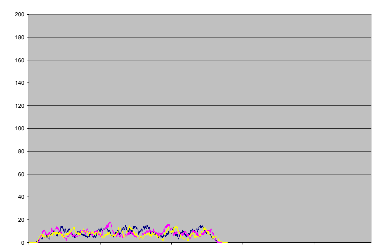
Link 3



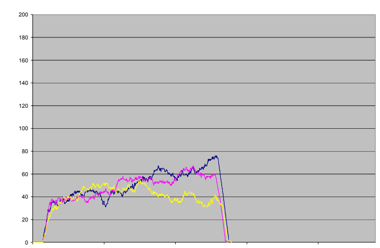
Link 11



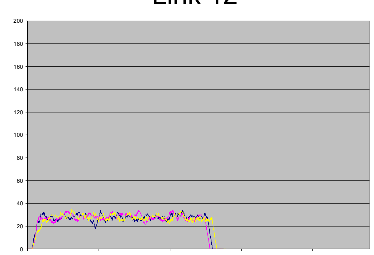
Link 10



Link 13

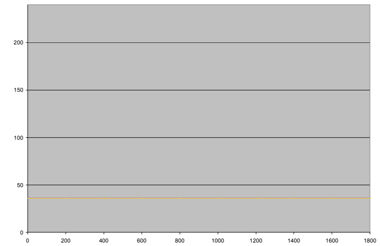


Link 12

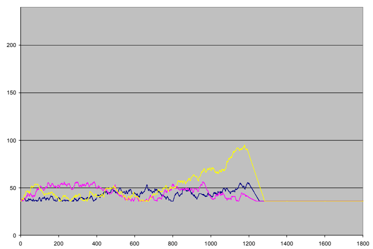


Link Volume Trajectories

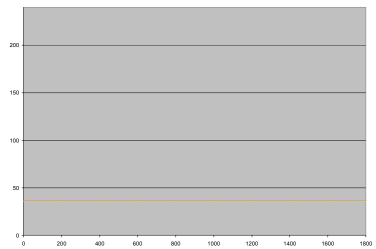
Link 4



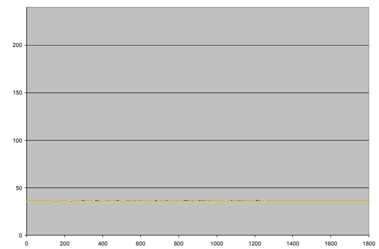
Link 1



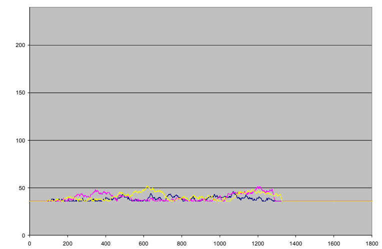
Link 5



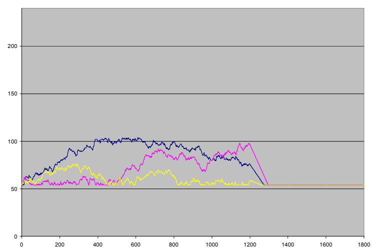
Link 8



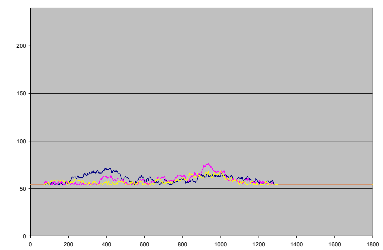
Link 6



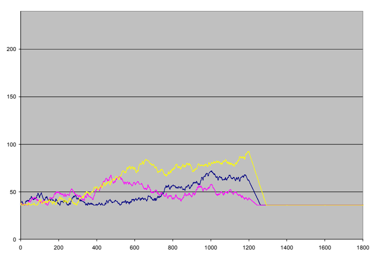
Link 2



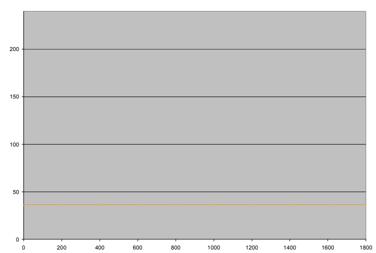
Link 9



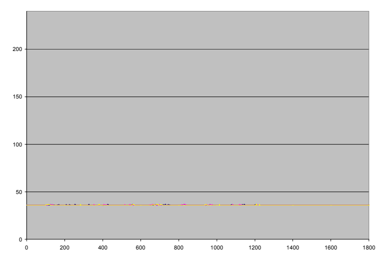
Link 3



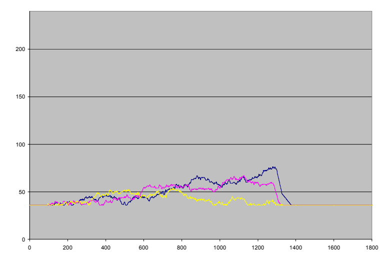
Link 11



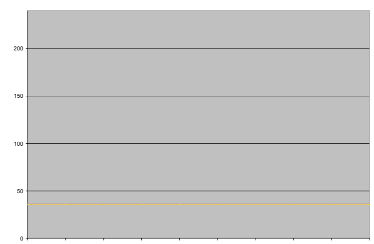
Link 10



Link 13

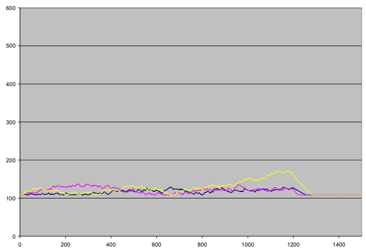


Link 12

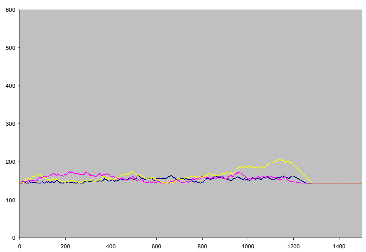


Link Time Trajectories

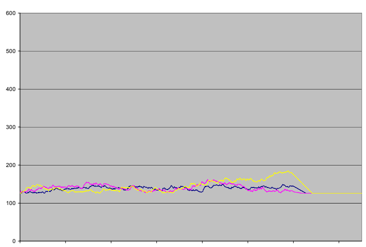
Path 0; links 1-4-6



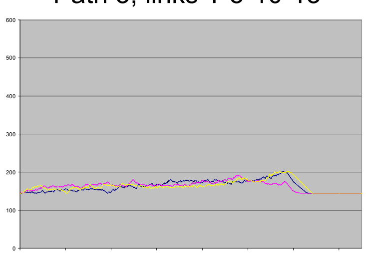
Path 1; links 1-5-8-6



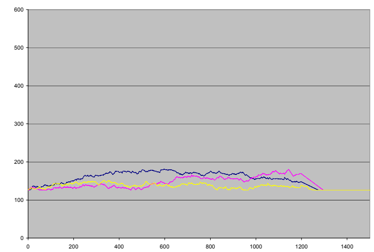
Path 2; links 1-5-9



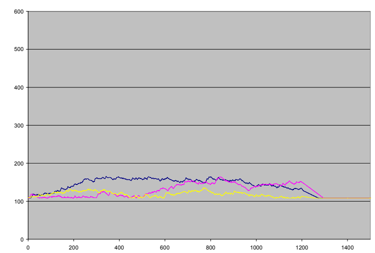
Path 3; links 1-5-10-13



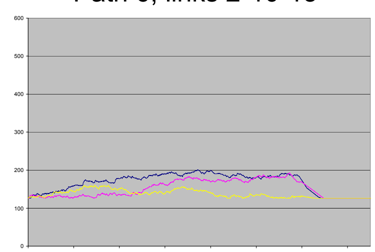
Path 5; links 2-8-6



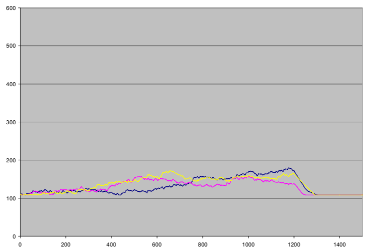
Path 4; links 2-9



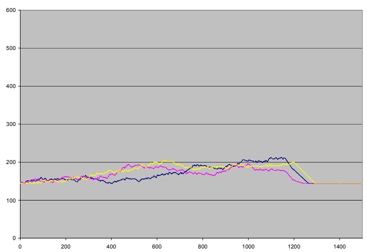
Path 6; links 2-10-13



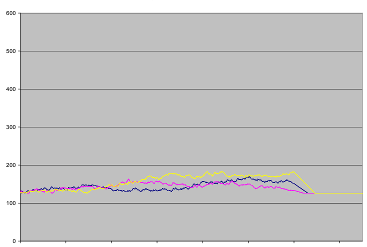
Path 7; links 3-12-13



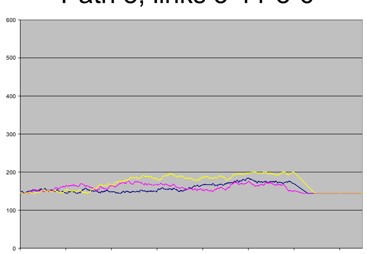
Path 10; links 3-11-10-13



Path 9; links 3-11-9

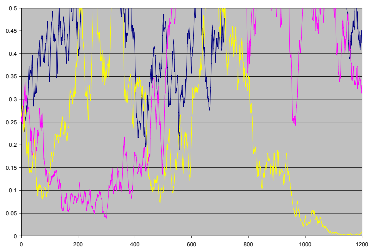


Path 8; links 3-11-8-6

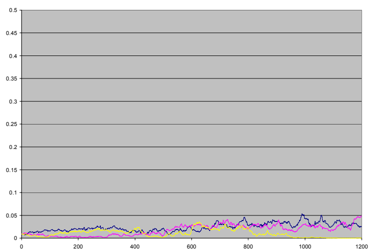


Path Time Trajectories

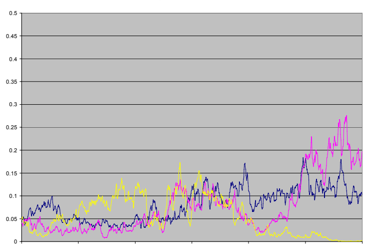
Path 0; links 1-4-6



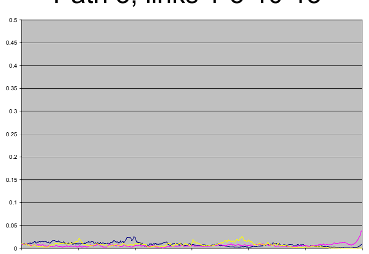
Path 1; links 1-5-8-6



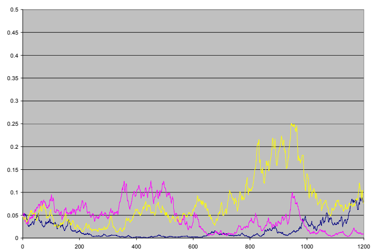
Path 2; links 1-5-9



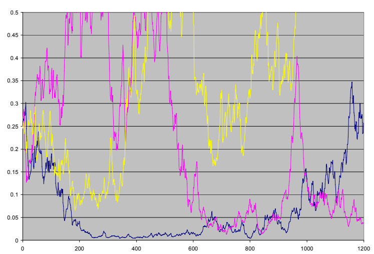
Path 3; links 1-5-10-13



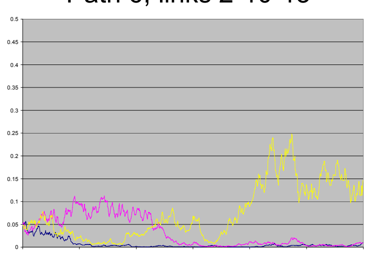
Path 5; links 2-8-6



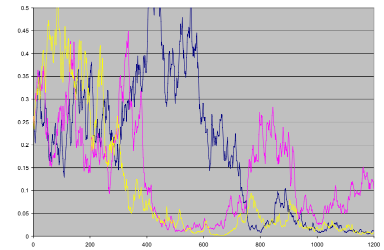
Path 4; links 2-9



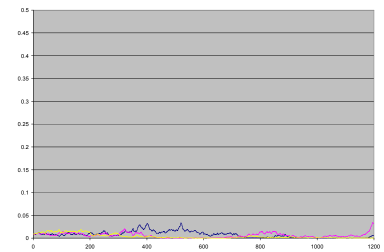
Path 6; links 2-10-13



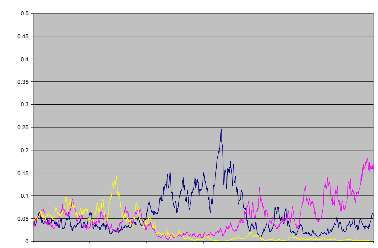
Path 7; links 3-12-13



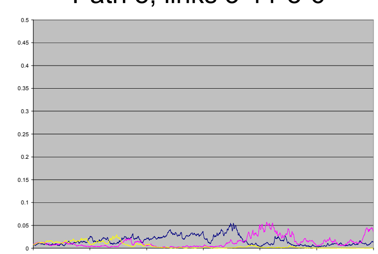
Path 10; links 3-11-10-13



Path 9; links 3-11-9

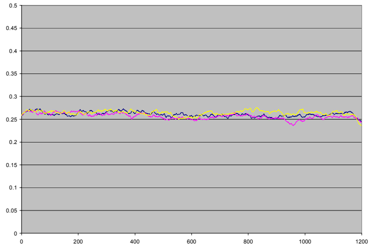


Path 8; links 3-11-8-6

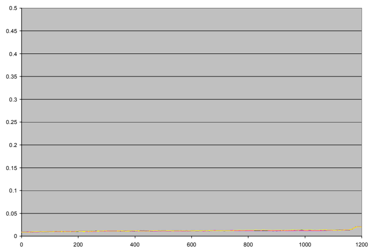


Output Pre-Trip Path Split Trajectories

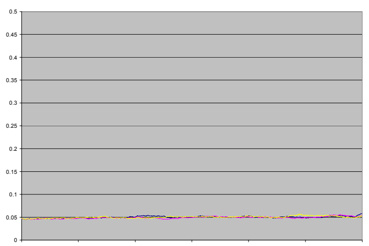
Path 0; links 1-4-6



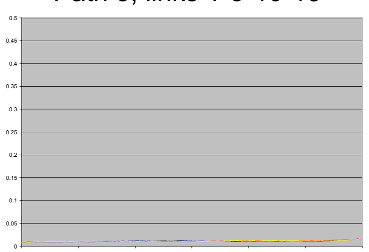
Path 1; links 1-5-8-6



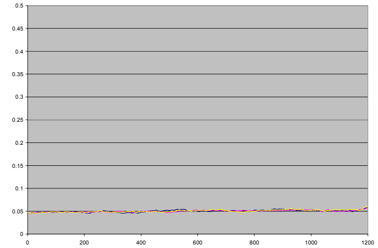
Path 2; links 1-5-9



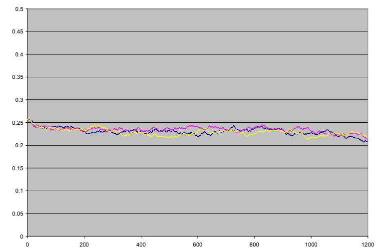
Path 3; links 1-5-10-13



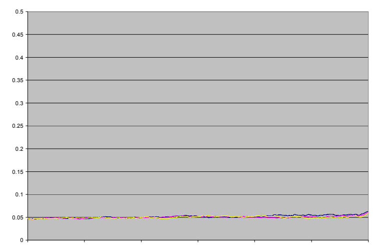
Path 5; links 2-8-6



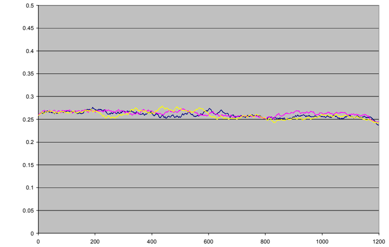
Path 4; links 2-9



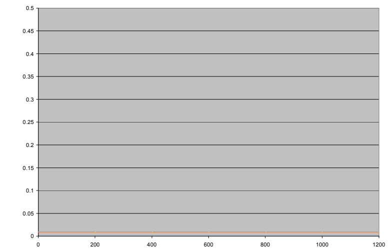
Path 6; links 2-10-13



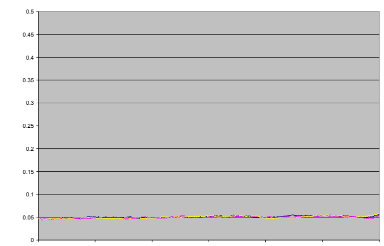
Path 7; links 3-12-13



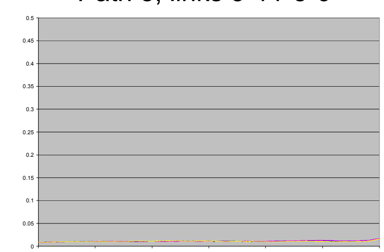
Path 10; links 3-11-10-13



Path 9; links 3-11-9

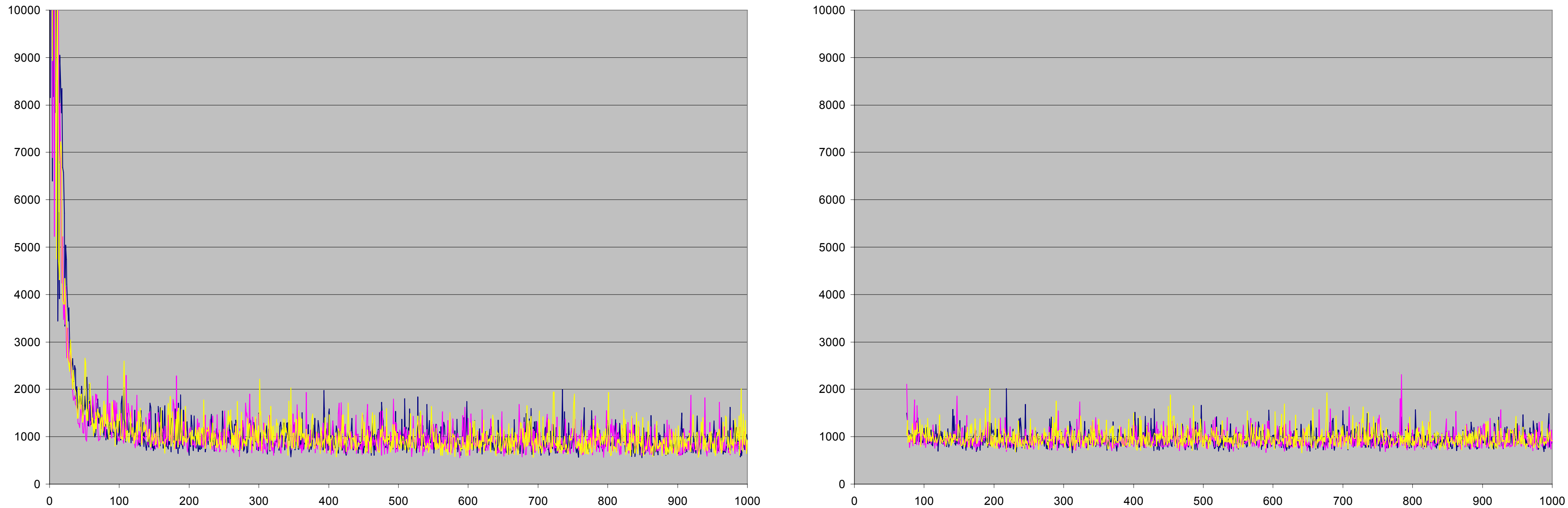


Path 8; links 3-11-8-6

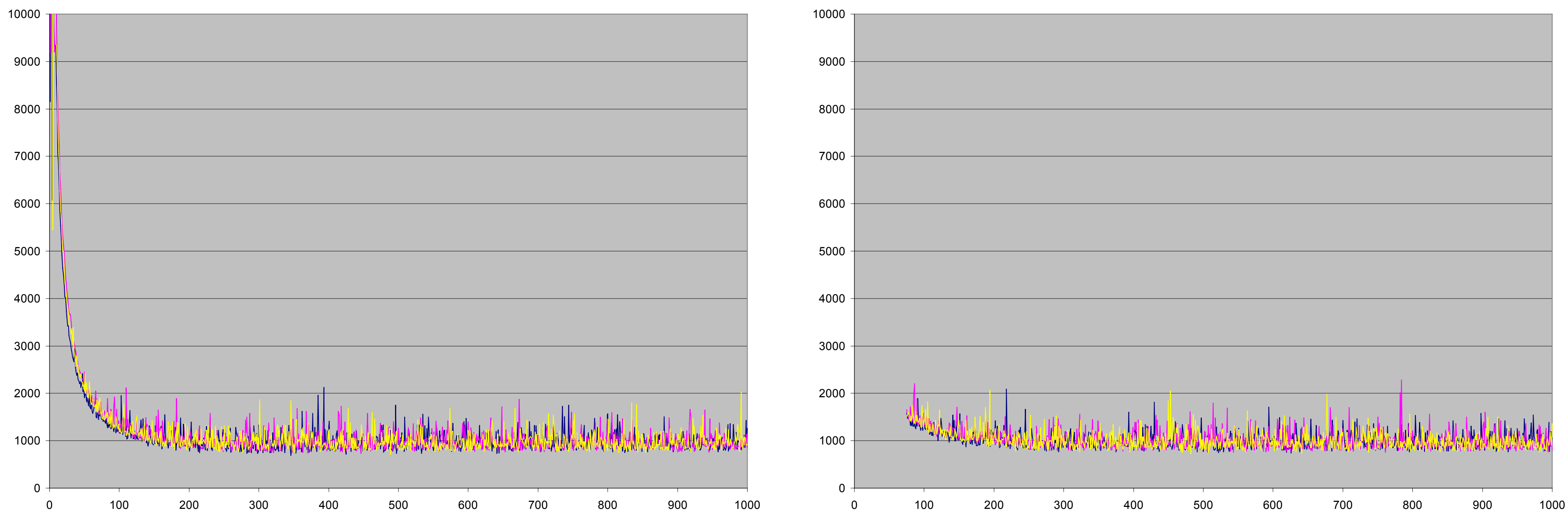


Input Pre-Trip Path Split Trajectories

Test Case 2_1_MSA_THEN_POLYAK

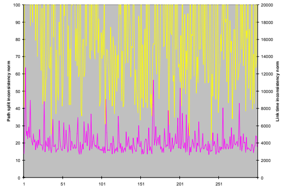
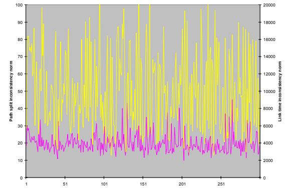
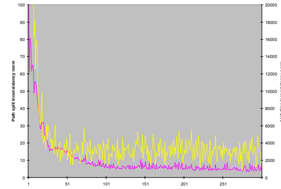
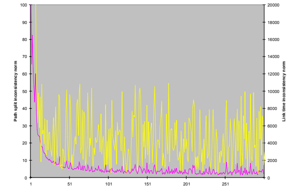
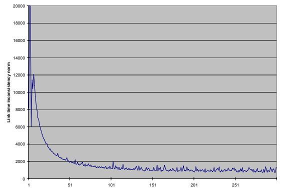
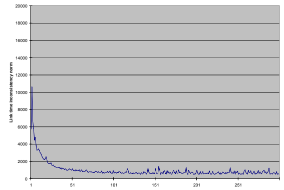
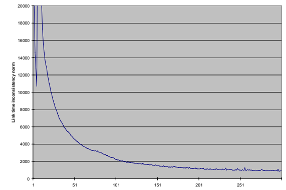
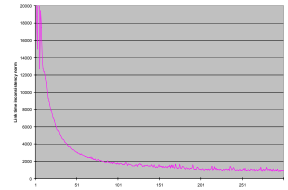


link time inconsistency norm, SDG – Polyak (Polyak then MSA)

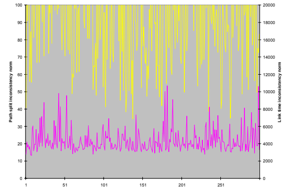
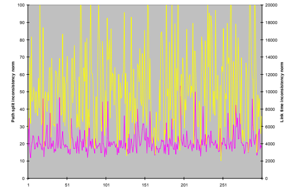
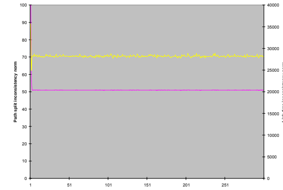
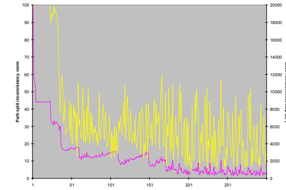
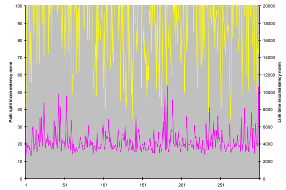
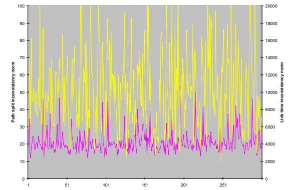
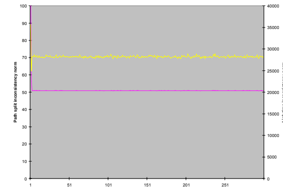
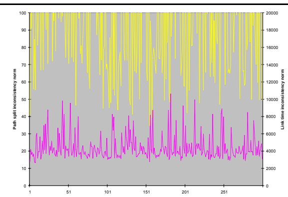
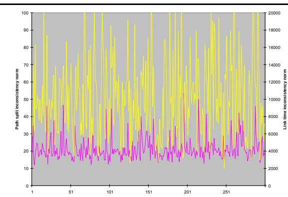
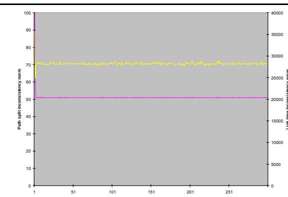
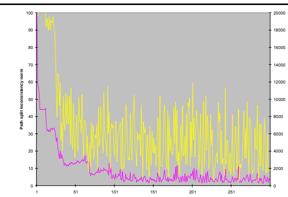
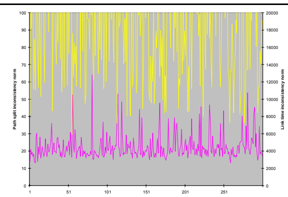
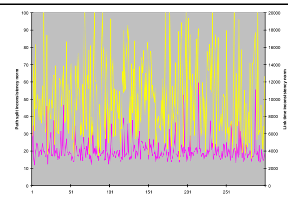
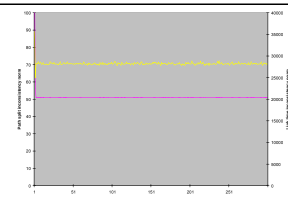
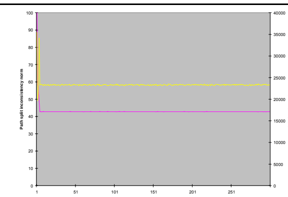
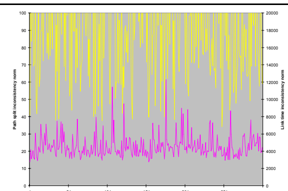
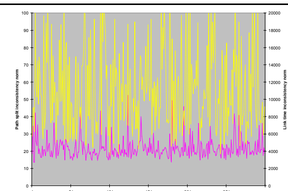
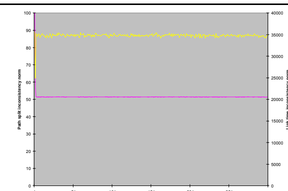
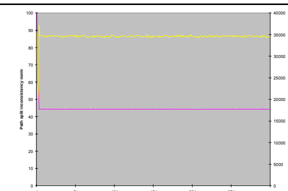


link time inconsistency norm, SDG – Polyak (MSA then Polyak)

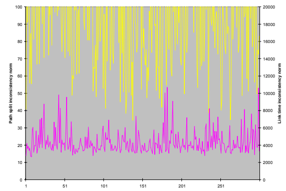
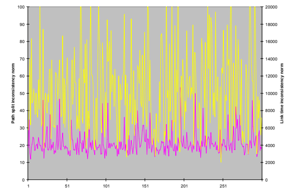
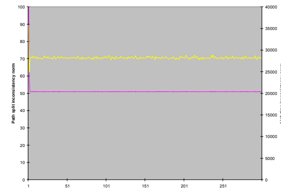
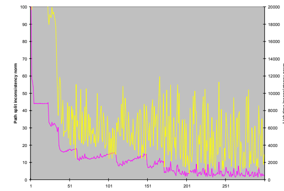
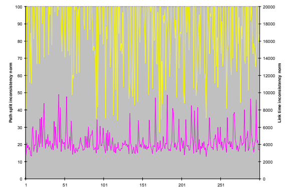
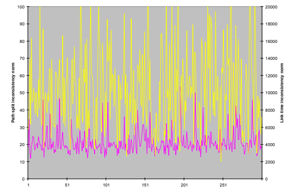
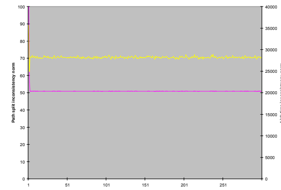
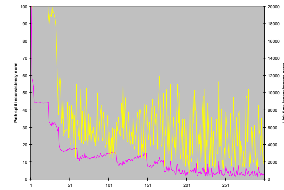
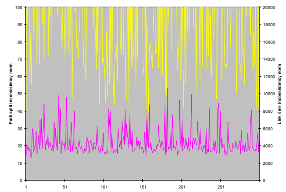
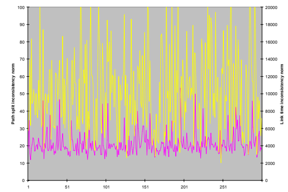
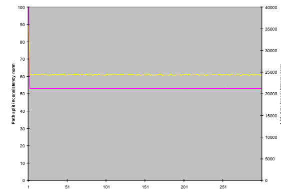
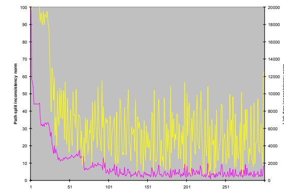
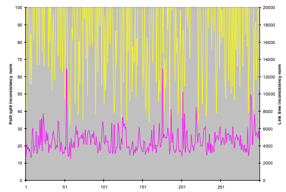
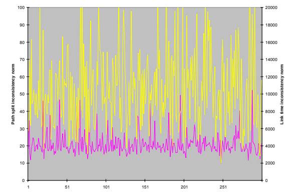
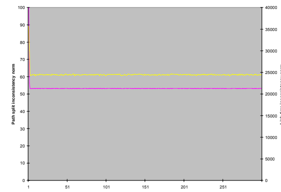
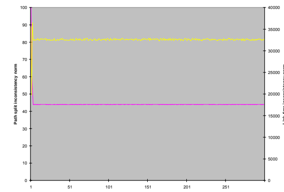
Test Case 2_2_POTENTIALS

Mapping	Step Size Computation Method	Network 2	Network 2, smaller	Network 3	Network 3, smaller
DGS	MSA				
SDG	MSA				

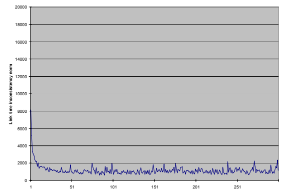
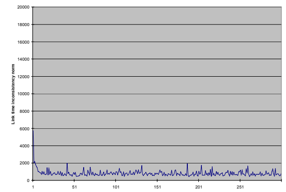
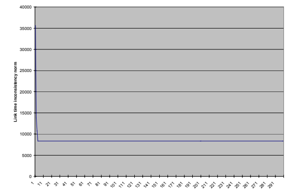
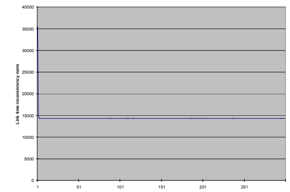
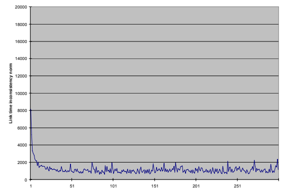
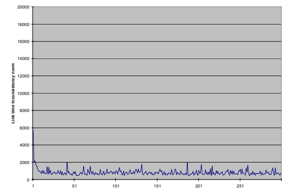
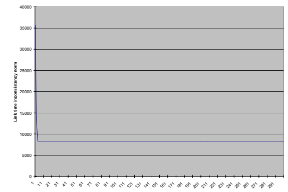
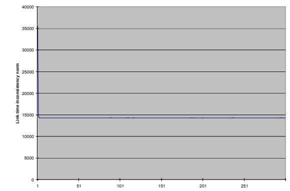
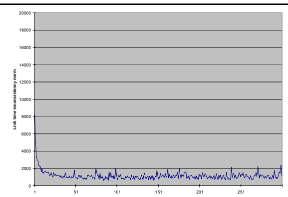
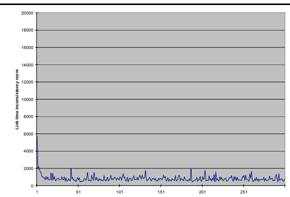
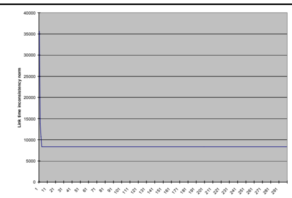
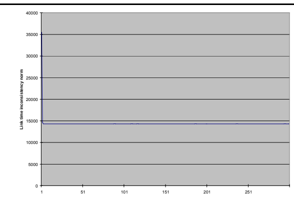
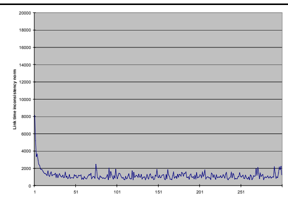
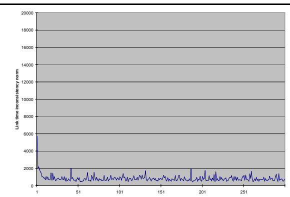
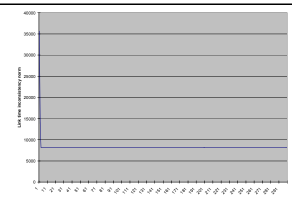
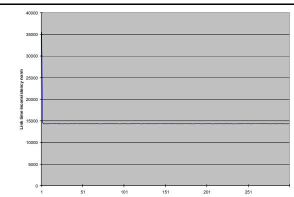
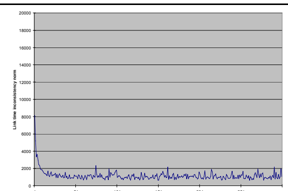
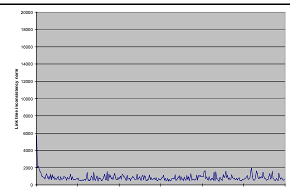
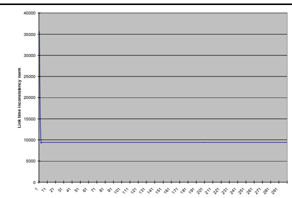
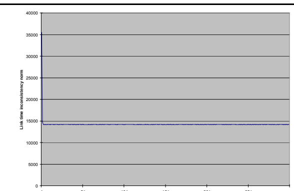
inconsistency norms

Mapping	Line Search Method, Potential Optimization	Network 2	Network 2, smaller	Network 3	Network 3, smaller
DGS	$\min_{\alpha} \left\| mapping\left(C^i(\alpha)\right) - C^i(\alpha) \right\|^2 - \beta\alpha(1-\alpha)\left\| mapping\left(C^i\right) - C^i \right\|^2$				
	with multiplier (β) = 0				
	with multiplier (β) = 0.001				
	with multiplier (β) = 0.01				
	with multiplier (β) = 0.1				
	with multiplier (β) = 1				

inconsistency norms

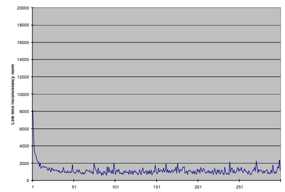
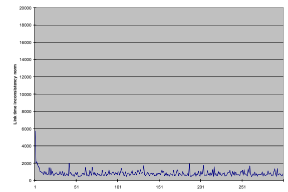
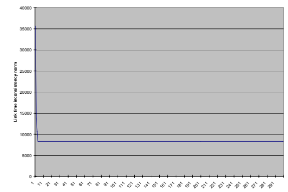
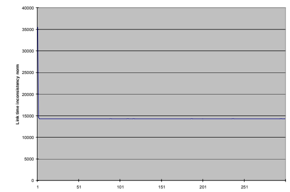
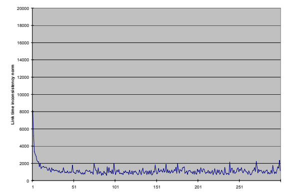
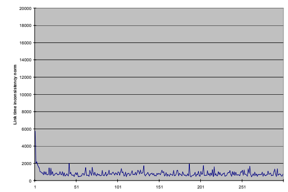
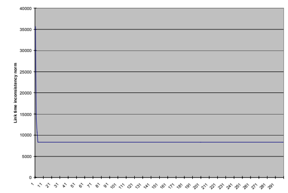
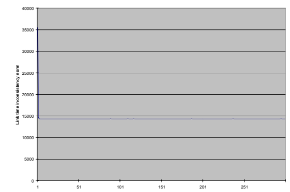
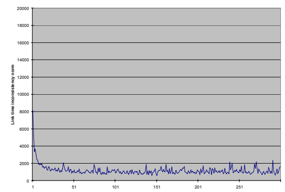
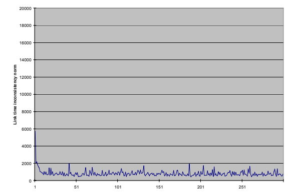
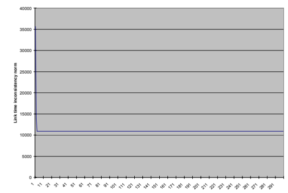
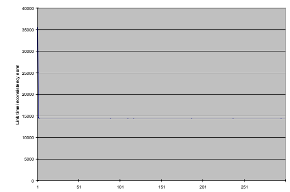
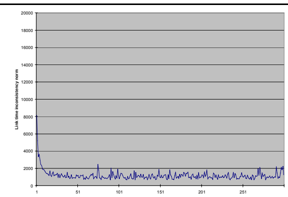
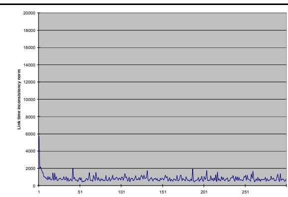
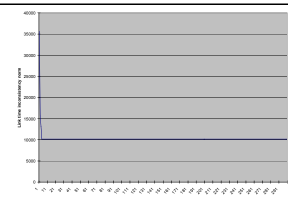
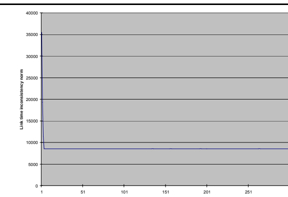
Mapping	Line Search Method, Potential Optimization	Network 2	Network 2, smaller	Network 3	Network 3, smaller
DGS	$\min\limits_{\alpha} \left\| mapping\left(C^i(\alpha)\right) - C^i(\alpha) \right\|^2 - \beta\alpha^2 \left\| mapping\left(C^i\right) - C^i \right\|^2$				
	with multiplier (β) = 0.001				
	with multiplier (β) = 0.01				
	with multiplier (β) = 0.1				
	with multiplier (β) = 1				

inconsistency norms

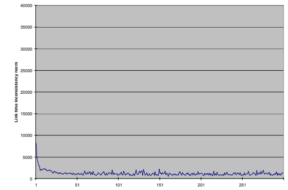
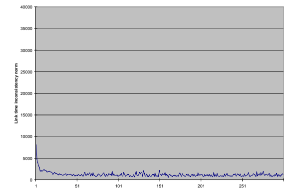
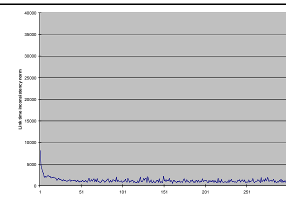
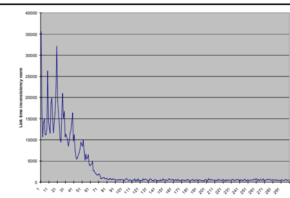
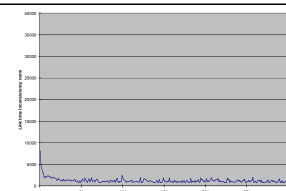
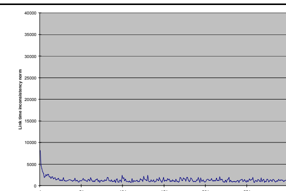
Mapping	Line Search Method, Potential Optimization	Network 2	Network 2, smaller	Network 3	Network 3, smaller
SDG	$\min\limits_{\alpha} \left\| mapping\left(C^i(\alpha)\right) - C^i(\alpha) \right\|^2 - \beta\alpha(1-\alpha)\left\| mapping\left(C^i\right) - C^i \right\|^2$				
	with multiplier (β) = 0				
	with multiplier (β) = 0.001				
	with multiplier (β) = 0.01				
	with multiplier (β) = 0.1				
	with multiplier (β) = 1				

inconsistency norms

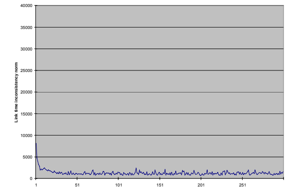
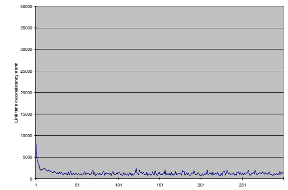
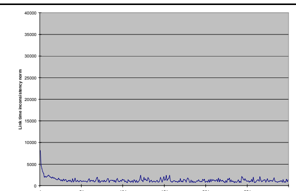
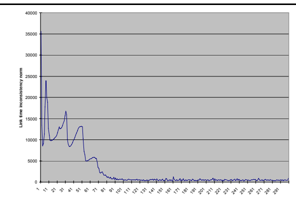
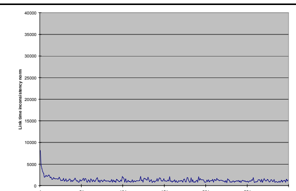
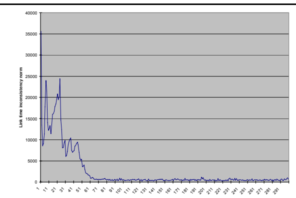
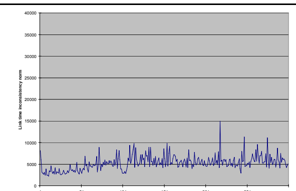
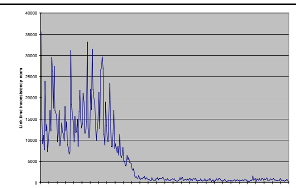
$$\min_{\alpha}\left\|mapping\left(C^{i}(\alpha)\right)-C^{i}(\alpha)\right\|^{2}-\beta\alpha^{2}\left\|mapping\left(C^{i}\right)-C^{i}\right\|^{2}$$

with multiplier (β) = 0.001				
with multiplier (β) = 0.01				
with multiplier (β) = 0.1				
with multiplier (β) = 1				

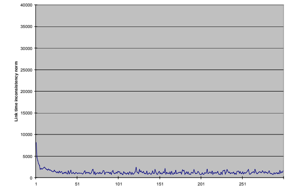
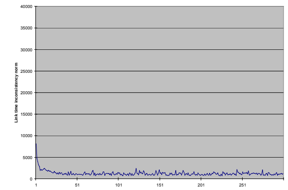
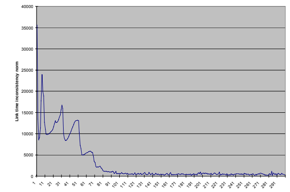
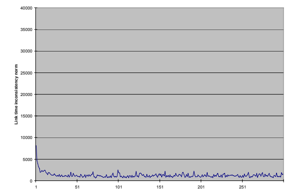
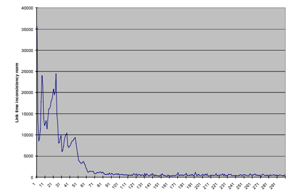
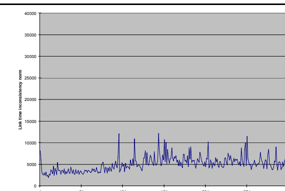
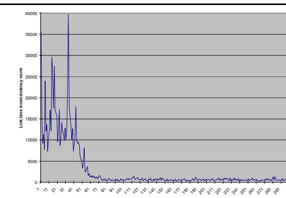
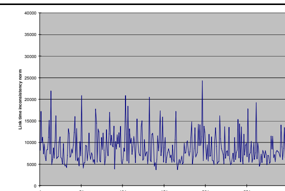
inconsistency norms

Mapping	Line Search Method, Potential Optimization	Network 2	Network 3
SDG	$\min\limits_{\alpha} \left\| mapping\left(C^i(\alpha)\right) - C^i(\alpha) \right\| - \beta\alpha(1-\alpha)\left\| mapping\left(C^i\right) - C^i \right\|$		
	with multiplier (β) = 0		
	with multiplier (β) = 0.001		
	with multiplier (β) = 0.01		
	with multiplier (β) = 0.1		
	with multiplier (β) = 1		

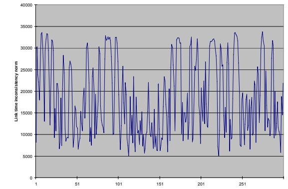
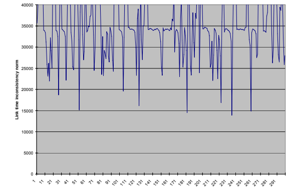
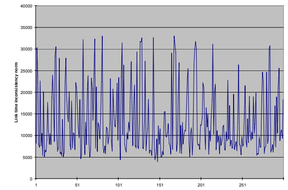
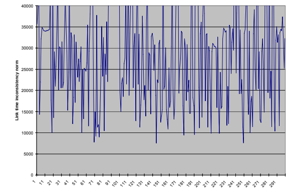
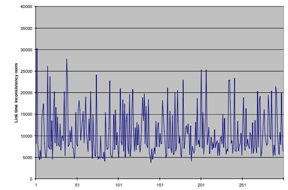
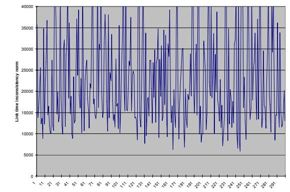
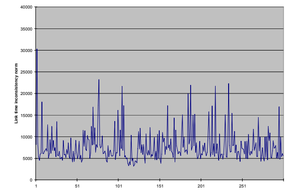
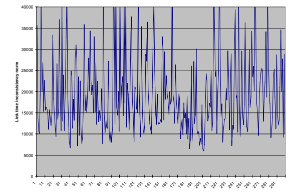
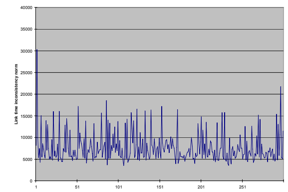
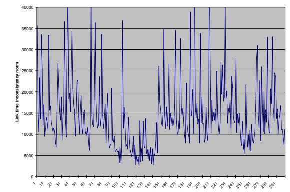
inconsistency norms

Mapping	Line Search Method, Potential Optimization	Network 2	Network 3
SDG	$\min\limits_{\alpha} \ln\left(\left\|mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right\|^2\right)-\beta\ln\left(\alpha(1-\alpha)\left\|mapping\left(C^i\right)-C^i\right\|^2\right)$		
	with multiplier (β) = 0		
	with multiplier (β) = 0.001		
	with multiplier (β) = 0.01		
	with multiplier (β) = 0.1		
	with multiplier (β) = 1		

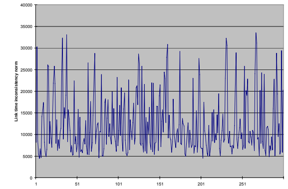
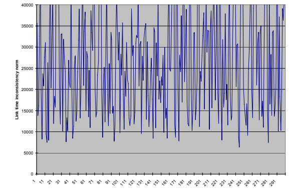
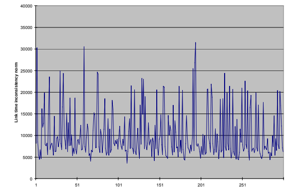
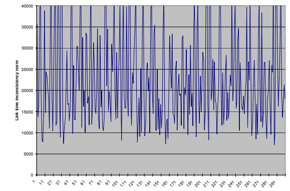
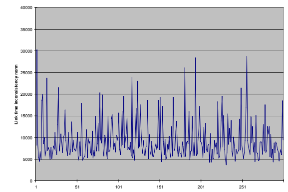
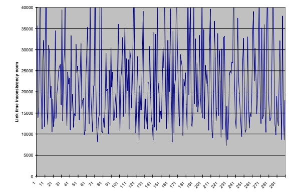
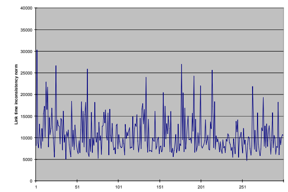
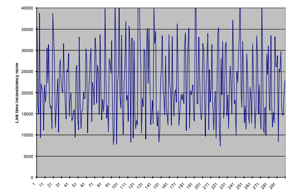
inconsistency norms

Mapping	Line Search Method, Potential Optimization	Network 2	Network 3
SDG	$\min_{\alpha} \ln\left(\left\Vert mapping\left(C^i(\alpha)\right) - C^i(\alpha)\right\Vert^2\right) - \beta \ln\left(\alpha \left\Vert mapping\left(C^i\right) - C^i\right\Vert\right) \ln\left((1-\alpha)\left\Vert mapping\left(C^i\right) - C^i\right\Vert\right)$		
	with multiplier (β) = 0		
	with multiplier (β) = 0.001		
	with multiplier (β) = 0.01		
	with multiplier (β) = 0.1		
	with multiplier (β) = 1		

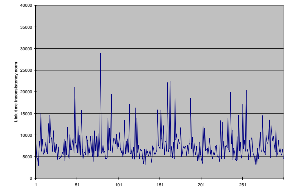
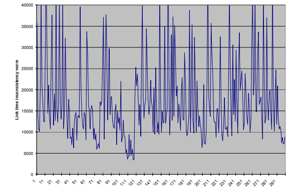
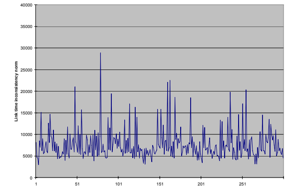
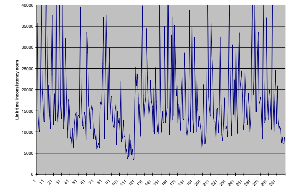
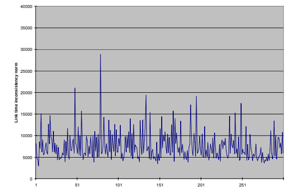
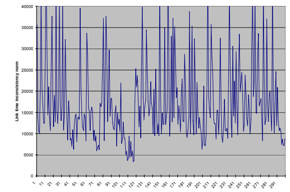
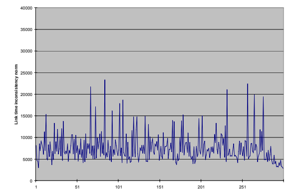
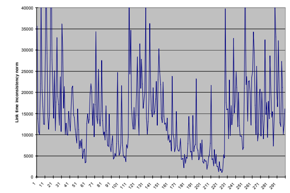
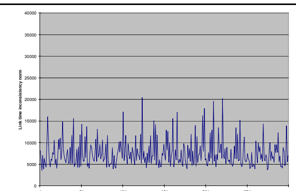
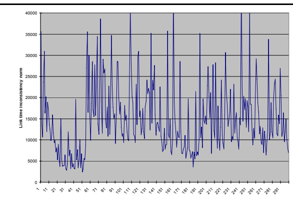
inconsistency norms

Mapping	Line Search Method, Potential Optimization	Network 2	Network 3
SDG	$\min\limits_{\alpha}\left(\left(mapping\left(C^{i}\right)-C^{i}\right)^{t}\left(mapping\left(C^{i}(\alpha)\right)-C^{i}(\alpha)\right)\right)^{2}-\beta\alpha(1-\alpha)\left\Vert mapping\left(C^{i}\right)-C^{i}\right\Vert^{4}$		
	with multiplier (β) = 0		
	with multiplier (β) = 0.001		
	with multiplier (β) = 0.01		
	with multiplier (β) = 0.1		
	with multiplier (β) = 1		

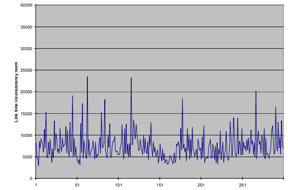
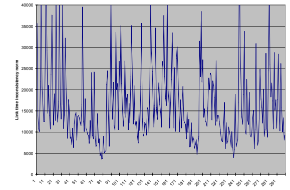
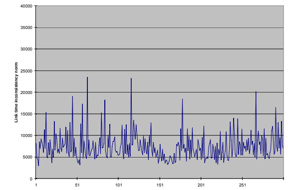
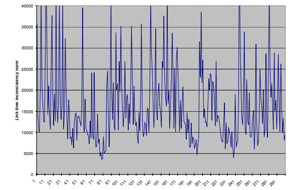
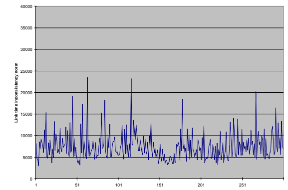
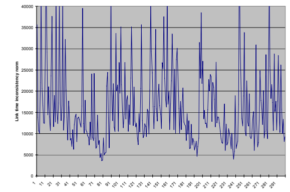
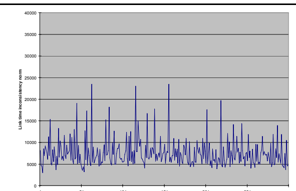
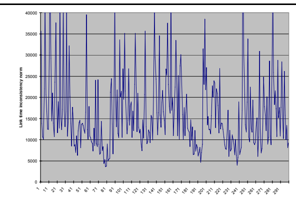
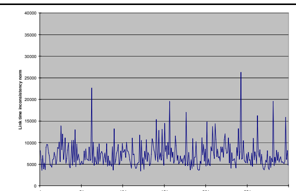
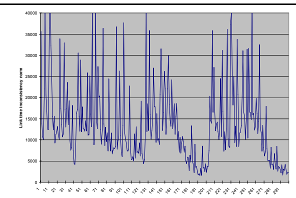
inconsistency norms

Mapping	Line Search Method, Potential Optimization	Network 2	Network 3
SDG	$\min\limits_{\alpha}\left(\left(mapping\left(C^{i}\right)-C^{i}\right)^{t}\left(mapping\left(C^{i}(\alpha)\right)-C^{i}(\alpha)\right)\right)^{2}-\beta\alpha^{2}\left\|mapping\left(C^{i}\right)-C^{i}\right\|^{4}$		
	with multiplier (β) = 0.001		
	with multiplier (β) = 0.01		
	with multiplier (β) = 0.1		
	with multiplier (β) = 1		

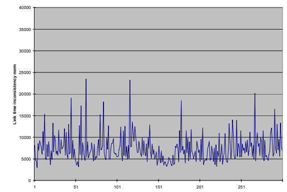
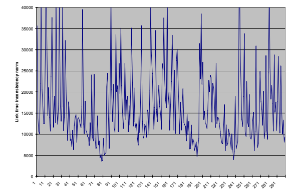
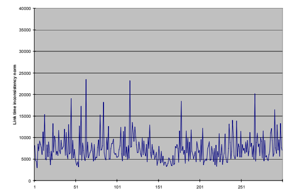
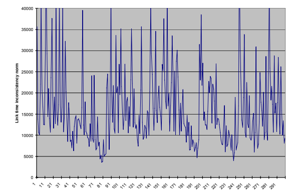
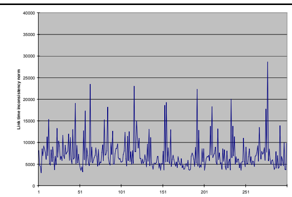
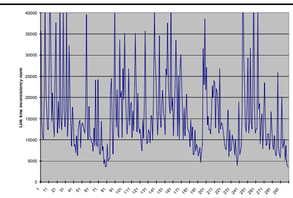
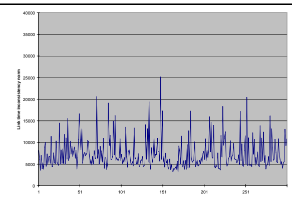
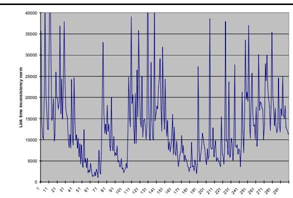
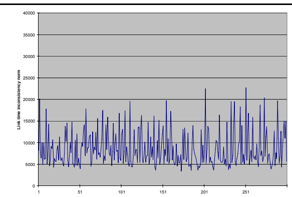
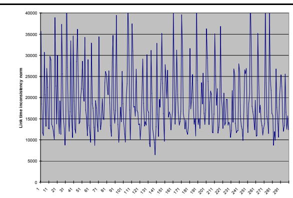
inconsistency norms

Mapping	Line Search Method, Potential Optimization	Network 2	Network 3
SDG	$\min_{\alpha}\left(mapping\left(C^i\right)-C^i\right)^t\left(mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right)-\beta\alpha(1-\alpha)\left\|mapping\left(C^i\right)-C^i\right\|^2$		
	with multiplier (β) = 0		
	with multiplier (β) = 0.001		
	with multiplier (β) = 0.01		
	with multiplier (β) = 0.1		
	with multiplier (β) = 1		

inconsistency norms

Mapping	Line Search Method, Potential Optimization	Network 2	Network 3
SDG	$\min_{\alpha} \ln\left(\left(\left(mapping\left(C^i\right)-C^i\right)^t\left(mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right)\right)^2\right)-\beta\ln\left(\alpha(1-\alpha)\left\|mapping\left(C^i\right)-C^i\right\|^4\right)$		
	with multiplier (β) = 0		
	with multiplier (β) = 0.001		
	with multiplier (β) = 0.01		
	with multiplier (β) = 0.1		
	with multiplier (β) = 1		

inconsistency norms

Mapping	Line Search Method, Potential Optimization	Network 2	Network 3
SDG	$\min_{\alpha} \ln\left(\left(\left(mapping\left(C^i\right)-C^i\right)^t\left(mapping\left(C^i(\alpha)\right)-C^i(\alpha)\right)\right)^2\right)$ $-\beta\ln\left(\alpha\left\|mapping\left(C^i\right)-C^i\right\|^2\right)\ln\left((1-\alpha)\left\|mapping\left(C^i\right)-C^i\right\|^2\right)$		
	with multiplier (β) = 0		
	with multiplier (β) = 0.001		
	with multiplier (β) = 0.01		
	with multiplier (β) = 0.1		
	with multiplier (β) = 1		

inconsistency norms

Results

Hardware Specification

Test cases were executed using a PC with the following specifications:

Platform: Microsoft Windows XP Home Edition
Processor: 1.80 GHz
RAM: 480 MB

Although the above configuration was sufficient to execute all of the test cases, it could not accommodate more refined scenarios of Test Case 2_2_ Potentials. Test cases of this scenario require a large number of evaluations in order to advance to the subsequent iteration. Using this hardware specification, the execution of such test cases was limited to 100 evaluations at each iteration.

Convergence

As expected during the study and implementation of this Anticipatory Route Guidance (ARG) framework, most test cases exhibit the convergence pattern. This is evident from the inconsistency graphs presented on pages 52, 63, 74, 79, 85, and 86- 98. In this regard, the contrast was clear between results of composite mapping $S \circ D \circ G(C^i)$ and those of $D \circ G \circ S(C^i)$, with composite mapping $S \circ D \circ G(C^i)$ resulting in better convergence patterns. The convergence resulting from $S \circ D \circ G(C^i)$ typically dropped faster to lower inconsistency norms than that of $D \circ G \circ S(C^i)$. Moreover, it stabilized better with less noise. It was also observed that the inconsistency norm never converged to zero. The research tested one hypothesis to address this concern. Page 63 is the result of a test case where stochasticity and rounding were minimized. As expected, the convergence was significantly lower than otherwise observed.

Method of Successive Averaging (MSA)

Among the test cases that investigate composite mapping $S \circ D \circ G(C^i)$, MSA (Method of Successive Averaging) algorithm consistently exhibited convergence. Moreover, compared to other averaging methods, this algorithm seems to be the most efficient in terms of computational effort. It was observed that test cases could complete in an average of 5 to 10 minutes using MSA algorithm compared to 60 to 90 minutes using Potential Optimization Line Search Algorithm. The inconsistency norm graphs produced by MSA algorithm are displayed on page 52, 63, 74, 85, and 86.

Polyak Iterate Averaging Algorithm

Page 85 displays the link time inconsistency norm diagrams of the Polyak Iterate Averaging algorithm. The research investigated two specific combinations of this averaging algorithm: (1) Polyak followed by MSA and (2) MSA followed by Polyak. Both cases resulted in similar convergence. One interesting observation from this comparison is that Polyak's convergence seems to drop the inconsistency norm faster to the stationary level.

Potential Optimization Line Search Algorithm

The research allocated a great deal of time and resources on evaluating the Potential Optimization Line Search Algorithm. The inconsistency norm graphs shown on pages 87 to 98 display the result of various test cases respective to different potentials and different multipliers (β). Again, these graphs demonstrate that composite mapping $S \circ D \circ G(C^i)$ resulted in better convergence patterns. The convergence resulting from $S \circ D \circ G(C^i)$ typically dropped faster and stabilized better with less noise. Moreover, the combination of composite mapping $S \circ D \circ G(C^i)$, a particular potential, and a particular multiplier (β) could result in much better convergence. These combinations are

Mapping	Potential	β	Network	Guidance	Path Splits	Flows	Times	Page
$S \circ D \circ G(C^i)$	$\min_{\alpha} \Vert S \circ D \circ G(C^i(\alpha)) - C^i(\alpha) \Vert - \beta\alpha(1-\alpha) \Vert S \circ D \circ G(C^i) - C^i \Vert$	0.01	link 102 - 200 at 900 v/h	100%	Aggregate	x10	x10	91
$S \circ D \circ G(C^i)$	$\min_{\alpha} \ln\left(\Vert mapping(C^i(\alpha)) - C^i(\alpha) \Vert^2 \right) - \beta \ln\left(\alpha(1-\alpha) \Vert mapping(C^i) - C^i \Vert^2 \right)$	0.01	link 102 - 200 at 900 v/h	100%	Aggregate	x10	x10	92
$S \circ D \circ G(C^i)$	$\min_{\alpha} \ln\left(\Vert mapping(C^i(\alpha)) - C^i(\alpha) \Vert^2 \right) - \beta \ln\left(\alpha(1-\alpha) \Vert mapping(C^i) - C^i \Vert^2 \right)$	0.1	link 102 - 200 at 900 v/h	100%	Aggregate	x10	x10	92
$S \circ D \circ G(C^i)$	$\min_{\alpha} \ln\left(\Vert mapping(C^i(\alpha)) - C^i(\alpha) \Vert^2 \right) - \beta \ln\left(\alpha(1-\alpha) \Vert mapping(C^i) - C^i \Vert^2 \right)$	1	link 102 - 200 at 900 v/h	100%	Aggregate	x10	x10	92
$S \circ D \circ G(C^i)$	$\min_{\alpha} \ln\left(\Vert mapping(C^i(\alpha)) - C^i(\alpha) \Vert^2 \right) - \beta \ln\left(\alpha \Vert mapping(C^i) - C^i \Vert\right) \ln\left((1-\alpha) \Vert mapping(C^i) - C^i \Vert\right)$	0.001	link 102 - 200 at 900 v/h	100%	Aggregate	x10	x10	93
$S \circ D \circ G(C^i)$	$\min_{\alpha} \ln\left(\Vert mapping(C^i(\alpha)) - C^i(\alpha) \Vert^2 \right) - \beta \ln\left(\alpha \Vert mapping(C^i) - C^i \Vert\right) \ln\left((1-\alpha) \Vert mapping(C^i) - C^i \Vert\right)$	0.01	link 102 - 200 at 900 v/h	100%	Aggregate	x10	x10	93
$S \circ D \circ G(C^i)$	$\min_{\alpha} \ln\left(\Vert mapping(C^i(\alpha)) - C^i(\alpha) \Vert^2 \right) - \beta \ln\left(\alpha \Vert mapping(C^i) - C^i \Vert\right) \ln\left((1-\alpha) \Vert mapping(C^i) - C^i \Vert\right)$	0.1	link 102 - 200 at 900 v/h	100%	Aggregate	x10	x10	93

Conclusion

In closing, we encourage further investigation using more scenarios of test cases and other averaging algorithms. One algorithm, for example, is an extension of the potential optimization line search algorithm where the optimization utilizes information from more than one past iteration. At the very least, further research should consider the scenarios of executing potential optimization line search algorithms using a larger number of iterations and/or evaluations for each iteration. This, of course, would require more sophisticated hardware, another practical improvement that can be accomplished following this research.

References

Bottom, J. A.: Consistent Anticipatory Route Guidance. PhD thesis, Center for Transportation Studies, Massachusetts Institute of Technology. 2000

Bottom, J. A., Kachani, S., Perakis, G.: The Anticipatory Route Guidance Problem: Formulations, Analysis and Computational Results. 2006

Magnanti, T. L. and Perakis, G.: Solving Variational Inequality and Fixed Point Problems by Line Searches and Potential Optimization. Mathematical Programming, 2004