

Short description of the parameters to LKH-3 (Version 3.0)

Parameters to LKH-3 are specified in a parameter file. All entries are of the form `<keyword> = <value>` (or `<keyword><whitespace><value>`), where `<keyword>` denotes an alphanumeric keyword and `<value>` denotes alphanumeric or numeric data. Keywords are not case sensitive.

The order of specifications in the file is arbitrary. The following specification is mandatory.

`PROBLEM_FILE = <string>`
Specifies the name of the problem file.

Additional control information may be supplied in the following format:

`ASCENT_CANDIDATES = <integer>`
The number of candidate edges to be associated with each node during the ascent. The candidate set is complemented such that every candidate edge is associated with both its two end nodes.
Default: 50.

`BACKBONE_TRIALS = <integer>`
The number of backbone trials in each run.
Default: 0.

`BACKTRACKING = { YES | NO }`
Specifies whether a backtracking k -opt move is to be used as the first move in a sequence of moves (where $k = \text{MOVE_TYPE}$).
Default: NO.

`BWTSP = <integer> <integer> [<integer>]`
Specifies the three parameters (B , Q , L) to a BWTSP instance.
 B : Number of black nodes.
 Q : Maximum number of white nodes on "black-to-black" paths.
 L : Maximum length of any "black-to-black" path.
Default: 0 0.

`CANDIDATE_FILE = <string>`
Specifies the name of a file to which the candidate sets are to be written. If, however, the file already exists, the candidate edges are read from the file. The first line of the file contains the dimension of the instance. Each of the following lines contains a node number, the number of the dad of the node in the minimum spanning tree (0, if the node has no dad), the number of candidate edges emanating from the node, followed by the candidate edges. For each candidate edge its end node number and alpha-value are given. It is possible to give more than one `CANDIDATE_FILE` specification. In this case the given files are read and the union of their candidate edges is used as candidate sets.

CANDIDATE_SET_TYPE = { ALPHA | DELAUNAY [PURE] |
NEAREST-NEIGHBOR | QUADRANT }

Specifies the candidate set type. ALPHA is LKH's default type. It is applicable in general. The other three types can only be used for instances given by coordinates. The optional suffix PURE for the DELAUNAY type specifies that only edges of the Delaunay graph are used as candidates.

Default: ALPHA.

COMMENT <string>

A comment.

<string>

A comment.

DEPOT = <integer>

Specifies the depot node.

Default: 1.

EDGE_FILE = <string>

Specifies the name of a file of candidate edges in Concorde format.

The first line of the file contains the number of nodes followed by the number of edges.

Each of the following lines contains the number of the two end nodes of an edge and its cost. The cost may be omitted. OBS: nodes are numbered from zero.

It is possible to give more than one EDGE_FILE specification. In this case the given files are read and the union of their candidate edges is used as candidate sets.

EOF

Terminates the input data. The entry is optional.

EXCESS = <real>

The maximum alpha-value allowed for any candidate edge is set to EXCESS times the absolute value of the lower bound of a solution tour (determined by the ascent).

Default: 1.0/DIMENSION.

EXTRA_CANDIDATES = <integer> [SYMMETRIC]

Number of extra candidate edges to be added to the candidate set of each node. Their candidate set type may be specified using the keyword EXTRA_CANDIDATE_SET_TYPE. The integer may be followed by the keyword SYMMETRIC, signifying that these extra candidate edges is to be complemented such that each of them is associated with both its two end nodes.

Default: 0.

EXTRA_CANDIDATE_SET_TYPE = { NEAREST-NEIGHBOR | QUADRANT }

The candidate set type of extra candidate edges.

Default: QUADRANT.

GAIN23 = [YES | NO]

Specifies whether the Gain23 function is used,
Default: YES.

GAIN_CRITERION = { YES | NO }

Specifies whether Lin and Kernighan's gain criterion is used.
Default: YES.

INITIAL_PERIOD = *<integer>*

The length of the first period in the ascent.
Default: DIMENSION/2 (but at least 100).

INITIAL_STEP_SIZE = *<integer>*

The initial step size used in the ascent.
Default: 1.

INITIAL_TOUR_ALGORITHM = { BORUVKA | CVRP | GREEDY | MOORE |
MTSP | NEAREST-NEIGHBOR |
QUICK-BORUVKA | SIERPINSKI | WALK }

Specifies the algorithm for obtaining an initial tour.
Default: WALK.

INITIAL_TOUR_FILE = *<string>*

Specifies the name of a file containing a tour to be used as the initial tour in the search. The tour is given by a list of integers giving the sequence in which the nodes are visited in the tour. The tour is terminated by a -1.

INITIAL_TOUR_FRACTION = *<real in [0;1]>*

Specifies the fraction of the initial tour to be constructed by means of INITIAL_TOUR_FILE edges.
Default: 1.0.

INPUT_TOUR_FILE = *<string>*

Specifies the name of a file containing a tour. The tour is used to limit the search (the last edge to be excluded in a non-gainful move must not belong to the tour). In addition, the Alpha field of its edges is set to zero. The tour is given by a list of integers giving the sequence in which the nodes are visited in the tour. The tour is terminated by a -1.

KICKS = *<integer>*

Specifies the number of times to "kick" a tour found by Lin-Kernighan. Each kick is a random *k*-swap kick. However, if KICKS is zero, then LKH's special kicking strategy, WALK, is used instead.
Default: 1.

KICK_TYPE = *<integer>*

Specifies the value of k for a random k -swap kick (an extension of the double-bridge move). If KICK_TYPE is zero, then the LKH's special kicking strategy, WALK, is used.

Default: 0.

MAKESPAN = { YES | NO }

Specifies if makespan optimization is to be used for a TSPTW instance.

Default: NO.

MAX_BREADTH = *<integer>*

Specifies the maximum number of candidate edges considered at each level of the search for a move.

Default: INT_MAX.

MAX_CANDIDATES = *<integer>* { SYMMETRIC }

The maximum number of candidate edges to be associated with each node. The integer may be followed by the keyword SYMMETRIC, signifying that the candidate set is to be complemented such that every candidate edge is associated with both its two end nodes. If MAX_CANDIDATES is zero the candidate sets are made up of the edges represented in the CANDIDATE_FILES, the INITIAL_TOUR_FILE, the INPUT_TOUR_FILE, the SUBPROBLEM_TOUR_FILE, and the MERGE_TOUR_FILES.

Default: 5.

MAX_SWAPS = *<integer>*

Specifies the maximum number of swaps (flips) allowed in any search for a tour improvement.

Default: DIMENSION.

MAX_TRIALS = *<integer>*

The maximum number of trials in each run.

Default: number of nodes (DIMENSION, given in the problem file).

MERGE_TOUR_FILE = *<string>*

Specifies the name of a tour to be merged. The edges of the tour are added to the candidate sets. It is possible to give more than two MERGE_TOUR_FILE specifications.

MOVE_TYPE = *<integer>* [SPECIAL]

Specifies the move type to be used in local search. An integer value $k \geq 2$ signifies that a sequential k -opt move is to be used. The specifier SPECIAL can be given in order to use LKH-3's specially designed moves. For this type of moves, k must be 3 or 5.

Default: 5.

MTSP_MIN_SIZE = *<integer>*

Specifies the minimum number of cities each salesman must visit in an MTSP or CVRP instance. If negative, its value is set to

$\text{DIMENSION}/(\text{ceil}(1.0 * \text{DIMENSION} / \text{MTSP_MAX_SIZE}) + 1)$

Default: 1.

MTSP_MAX_SIZE = *<integer>*

Specifies the maximum number of cities each salesman may visit in an MTSP or CVRP instance.

Default: DIMENSION - 1.

MTSP_OBJECTIVE = [MINMAX | MINMAX_SIZE | MINSUM]

Specifies the **objective function** type for an MTSP or CVRP instance.

MINMAX - Minimize the length of the longest route.

MINMAX_SIZE - Minimize the size of the largest route.

MINSUM - Minimize the length of the overall tour.

All its routes must satisfy the MTSP_MIN_SIZE and MTSP_MAX_SIZE constraints.

MTSP_SOLUTION_FILE = *<string>*

Specifies the name of a file where the solution of an MTSP (or VRP) instance is to be written. The character \$ in the name has a special meaning. All occurrences are replaced by the cost of the tour.

NONSEQUENTIAL_MOVE_TYPE = *<integer>*

Specifies the nonsequential move type to be used. A value $K \geq 4$ signifies that attempts are made to improve a tour by nonsequential k -opt moves where $4 \leq k \leq K$. Note, however, that the effect depends on the specifications of PATCHING_A and PATCHING_B.

Default: MOVE_TYPE + PATCHING_A + PATCHING_B - 1.

OUTPUT_TOUR_FILE = *<string>*

Specifies the name of a file where the best tour is to be written. Each time a trial has produced a new best tour, the tour is written to this file. The character \$ in the name has a special meaning. All occurrences are replaced by the cost of the tour.

OPTIMUM = *<integer>*

Known optimal tour length. If STOP_AT_OPTIMUM is YES, a run will be terminated if the tour length becomes equal to this value.

Default: -LLONG_MIN.

PATCHING_A = *<integer>* [RESTRICTED | EXTENDED]

The maximum number of disjoint alternating cycles to be used for patching. An attempt to patch cycles is made if the corresponding non-sequential move is gainful. The integer may be followed by the keyword RESTRICTED or EXTENDED. The keyword RESTRICTED signifies that gainful moves are only considered if all its inclusion edges are candidate edges. The keyword EXTENDED signifies that the non-sequential move need not be gainful if only all its inclusion edges are candidate edges.

Default: 1.

PATCHING_C = *<integer>* [RESTRICTED | EXTENDED]

The maximum number of disjoint cycles to be patched in an attempt to find a feasible and gainful move. An attempt to patch cycles is made if the corresponding non-sequential move is gainful. The integer may be followed by the keyword RESTRICTED or EXTENDED. The keyword RESTRICTED signifies that gainful moves are only considered if all its inclusion edges are candidate edges. The keyword EXTENDED signifies that the non-sequential move need not be gainful if only all its inclusion edges are candidate edges.

Default: 0.

PI_FILE = *<string>*

Specifies the name of a file to which penalties (Pi-values determined by the ascent) are to be written. If the file already exists, the penalties are read from the file, and the ascent is skipped. The first line of the file contains the number of nodes. Each of the following lines is of the form

<integer> *<integer>*

where the first integer is a node number, and the second integer is the Pi-value associated with the node. The file name "0" represents a file with all Pi-values equal to zero.

POPULATION_SIZE = *<integer>*

Specifies the maximum size of the population in LKH's genetic algorithm. Tours found by the first POPULATION_SIZE runs constitute an initial population of tours. In each of the remaining runs two tours (parents) from the current population is recombined into a new tour (child) using a variant of the Edge Recombination Crossover (ERX). The parents are chosen with random linear bias towards the best members of the population. The child is used as initial tour for the next run. If this run produces a tour better than the worst tour of the population, then the resulting tour replaces the worst tour. Premature convergence is avoided by requiring that all tours in the population have different costs.

Default: 0.

PRECISION = *<integer>*

The internal precision in the representation of transformed distances:

$$d[i][j] = \text{PRECISION} * c[i][j] + pi[i] + pi[j],$$

where $d[i][j]$, $c[i][j]$, $pi[i]$ and $pi[j]$ are all integral.

Default: 100 (which corresponds to 2 decimal places).

RESTRICTED_SEARCH = [YES | NO]

Specifies whether the following search pruning technique is used: The first edge to be broken in a move must not belong to the currently best solution tour. When no solution tour is known, it must not belong to the minimum spanning 1-tree.

Default: YES.

RUNS = *<integer>*

The total number of runs.

Default: 10.

SALESMEN = *<integer>*

Specifies the number of salesmen in an MTSP or VRP instance.

Default: 1.

SCALE = *<integer>*

Scale factor for Euclidean and ATT instances.

Default: 1.

SEED = *<integer>*

Specifies the initial seed for random number generation. If zero, the seed is derived from the system clock.

Default: 1.

SINTEF_SOLUTION_FILE = *<string>*

Specifies the name of a file where the solution of an MTSP or VRP instance is to be written. The solution is written in SINTEF format. The character \$ in the name has a special meaning. All occurrences are replaced by the cost of the tour.

SPECIAL

Equivalent to the following non-default parameter specifications:

GAIN23 = NO

KICKS = 1

KICKTYPE = 4

MAX_SWAPS = 0

MOVE_TYPE = 5 SPECIAL

POPULATION_SIZE = 10

STOP_AT_OPTIMUM = [YES | NO]

Specifies whether a run is stopped, if the tour length becomes equal to OPTIMUM.

Default: YES.

SUBGRADIENT = [YES | NO]

Specifies whether the pi-values should be determined by subgradient optimization.

Default: YES.

SUBPROBLEM_SIZE = *<integer>* [DELAUNAY | KARP | K-CENTER | K-MEANS |
MOORE | SIERPINSKI]
[BORDERS] [COMPRESSED]

The number of nodes in a division of the original problem into subproblems. The division is made according to the tour given by SUBPROBLEM_TOUR_FILE. The value 0 signifies that no division is made. By default, the subproblems are determined by subdividing the tour into segments of equal size. However, the integer may be followed by DELAUNAY, KARP, K-CENTER, K-MEANS, MOORE, ROHE, or SIERPINSKI. DELAUNAY signifies that a Delaunay partitioning scheme is used, KARP that Karp's partitioning scheme is used, K-CENTER that a partitioning scheme based on K-center clustering is used, K-MEANS that a partitioning scheme based on K-means clustering is used, ROHE that André Rohe's random rectangle partitioning scheme is used, and MOORE or SIERPINSKI that a partitioning scheme based on either a Moore or a Sierpinski space-filling curve is used. The BORDERS specification signifies that the subproblems along the borders between subproblems are to be solved too.

The COMPRESSED specification signifies that each subproblem is compressed by removing from the problem all nodes with two incident subproblem tour edges that belong to all tours to be merged (at least two MERGE_TOUR_FILES should be given).

Default: 0.

SUBPROBLEM_TOUR_FILE = *<string>*

Specifies the name of a file containing a tour to be used for dividing the original problem into subproblems. The approximate number of nodes in each is given by SUBPROBLEM_SIZE. The tour is given by a list of integers giving the sequence in which the nodes are visited in the tour. The tour is terminated by a -1.

SUBSEQUENT_MOVE_TYPE = *<integer>* [SPECIAL]

Specifies the move type to be used for all moves following the first move in a chain of moves. The value $k \geq 2$ signifies that a k -opt move is to be used. The value 0 signifies that all moves are of the same type ($k = \text{MOVE_TYPE}$). The specifier SPECIAL can be given in order to use LKH-3's specially designed moves. For this type of moves, k must be 3 or 5.

Default: 0.

SUBSEQUENT_PATCHING = { YES | NO }

Specifies whether patching is used for moves following the first move in a sequence of moves.

Default: YES.

TIME_LIMIT = *<real>*

Specifies a time limit in seconds for each run.

Default: DBL_MAX.

TOUR_FILE = *<string>*

Specifies the name of a file to which the best tour is to be written. The character \$ in the name has a special meaning. All occurrences are replaced by the cost of the tour.

TRACE_LEVEL = *<integer>*

Specifies the level of detail of the output given during the solution process. The value 0 signifies a minimum amount of output. The higher the value is the more information is given.

Default: 1.

VEHICLES = *<integer>*

Specifies the number of vehicles/salesmen in an MTSP or VRP instance.

Default: 1.

List of abbreviations

A string value may be abbreviated to the first few letters of the string, if that abbreviation is unambiguous.

<i>Value</i>	<i>Abbreviation</i>
ALPHA	A
BORDERS	B
BORUVKA	B
COMPRESSED	C
CVRP	C
DELAUNAY	D
EXTENDED	E
GREEDY	G
KARP	KA
K-CENTER	K-C
K-MEANS	K-M
MOORE	M
NEAREST-NEIGHBOR	N
NO	N
PURE	P
QUADRANT	Q
QUICK-BORUVKA	Q
RESTRICTED	R
ROHE	R
SIERPINSKI	SI
SOP	SO
SPECIAL	S
SYMMETRIC	S
WALK	W
YES	Y