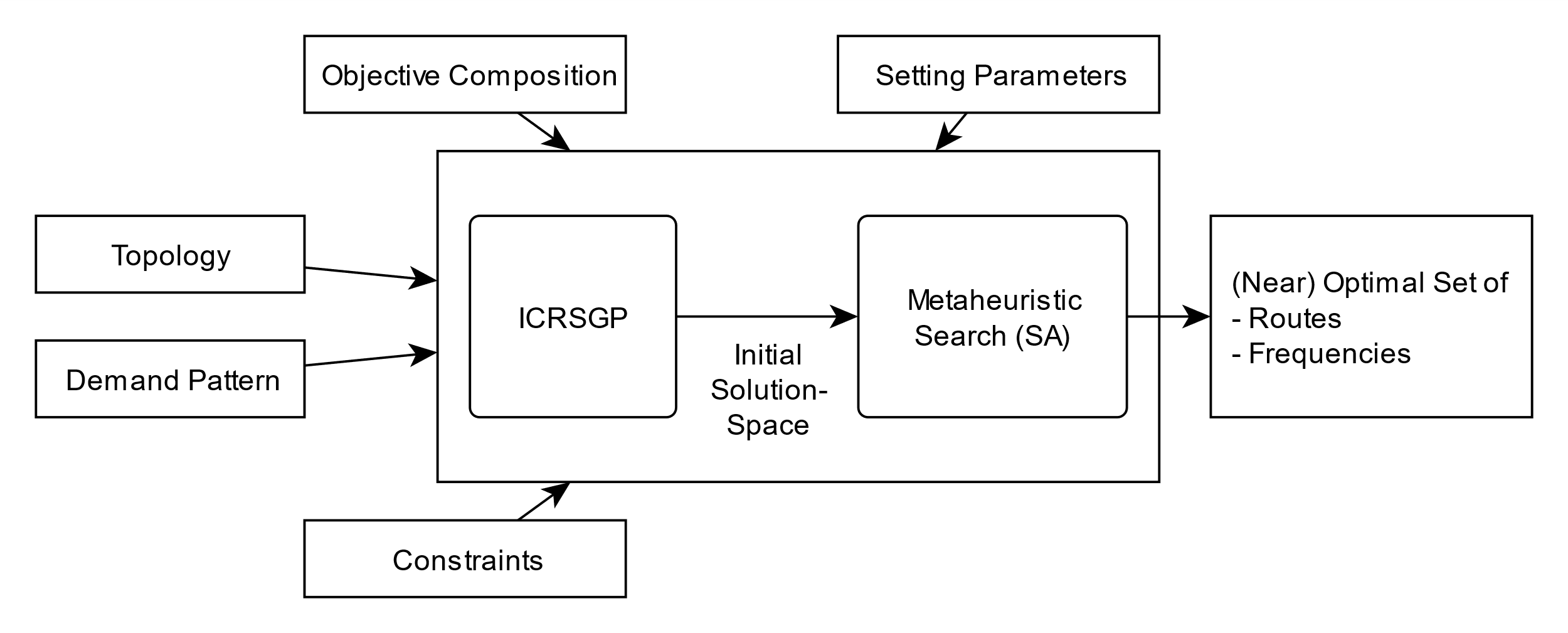
In order to verify the validity of the aforementioned methods of creating input data for conduction of theoretical experiments, we will introduce a test-algorithm from the area of network planning.

# Route Generation and Scheduling using Simulated Annealing



In the following parts we will be introducing an algorithm for the bus transit route network design problem (BTRNDP) [] mainly based on a proposition in […..]. In reference to our practical experience and adapting to the given circumstances of the experimental environment, we issued modifications, especially simplifications, to the proposition.

## General Behavior

As shown in Fig1, given the characteristics of a transportation network our test-algorithm will generate a (near-) optimal set of routes for the vehicles as well as the respective frequencies. The test-algorithm implements the optimization process using a metaheuristic approach inspired by the concept of “Simulated Annealing”. Besides the aforementioned characteristics of the transportation network the result will also depend on certain setting parameters for the computation process given by the user.

## Transport Network Input Data

First, we will provide a more detailed description of the data to be fed into the test-algorithm.

The transportation network is captured by an undirected graph containing nodes representing distribution point of the network and edges representing feasible connections between these nodes. In case of the bus transit route network, the nodes can be interpreted as the relevant stops accessible by the vehicles.

In the following context we will assume a network consisting of nodes and

edges.

### Topology

The topology of a given transportation network is expected to be in the form a matrix two-dimensional matrix, .

The categorizing between accessible nodes and non-feasible connection has a great will have a great influence on the generation of routes. Pairs of nodes that cannot to connected in a feasible manner will decrease the solution space for result of optimization. Possible criteria for the categorization may be a limited span for accepted travelling times and the enforcement onto existing routes.

### Demand Pattern

The most influential characteristic of a transportation network regarding our test-algorithm is the description of flow between each pair of nodes for a certain span of time. This aspect directly correlates with demand pattern.

The demand pattern of the network will be captured by a two-dimensional matrix, ., also called “Origin-Destination-Matrix”.

### Constraints

Depending on the properties and limitations regarding the transportations network as well as policy rules, the solution space must be adapted to fit those restrictions. For the purpose of this test-algorithm and the theoretical environment, we will consider a subset of proposed constraints in […].

The constraints mentioned above not only influence the composition of the solution space, but they also greatly influence the computational runtime characteristics of the test-algorithm. A further clarification will be offered together with a detailed explanation of the structure of the algorithm.

### Objective Composition

The metaheuristic search procedure is dependent on the comparability of test-solutions regarding their suitability for the demands of the user. We define the value of a test-solution by establishing an objective function build as a composition of multiple unique objectives. For this purpose, we will use a simplified cost-function inspired by the proposal in […], which will associate each solution with a score.

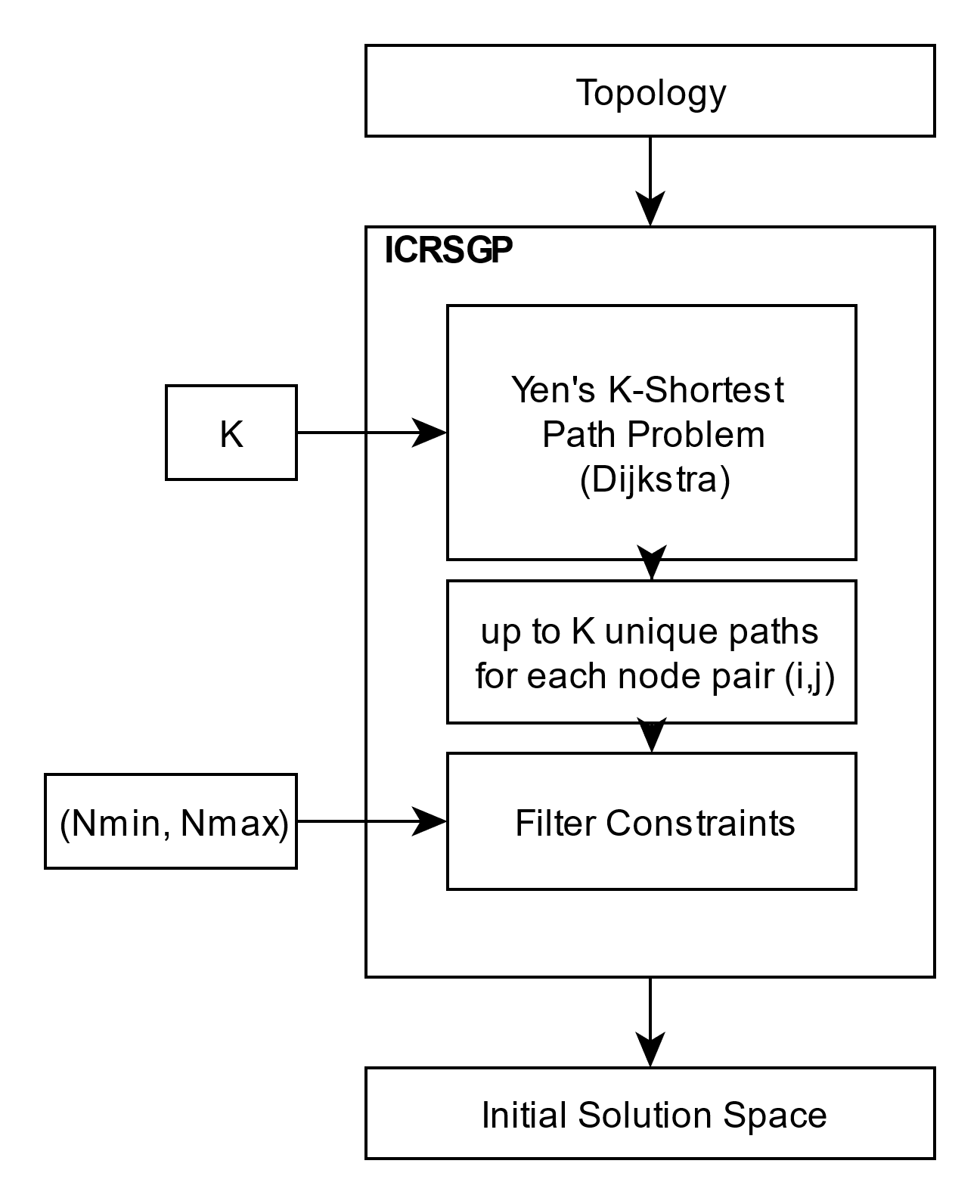
The User Score measures the level of comfort for the passengers trying to travel their respective destinations. The User Score differentiates between passengers who can travel without transfers and those who are forced to. Transfers will be penalized by the factor user-defined factor . The total user score will be weighted by the value given by the user.

The Operation Score measures the size of the homogenous set of needed vehicles to accommodate the demand of the test solution. This value will be weighted by the factor as defined by the user.

Lastly, we will also consider the amount of demand not fulfilled by the current solution. This value is weighted by the value .

Additionally, we can conclude a certain dependency between the User Score and the unfulfilled demand. This statement can be derived from the fact, that unfilled demand may indicate to pairs of nodes that are virtually isolated from each other.

## ICRSGP



As described in [1] the initial candidate route set generation procedure (ICRSGP) will generate a set of paths between each pair of nodes in the network and offer an Initial Solution Space that will serve as the foundation of the metaheuristic search procedure.

### Yen’s K-Shortest Path Algorithm

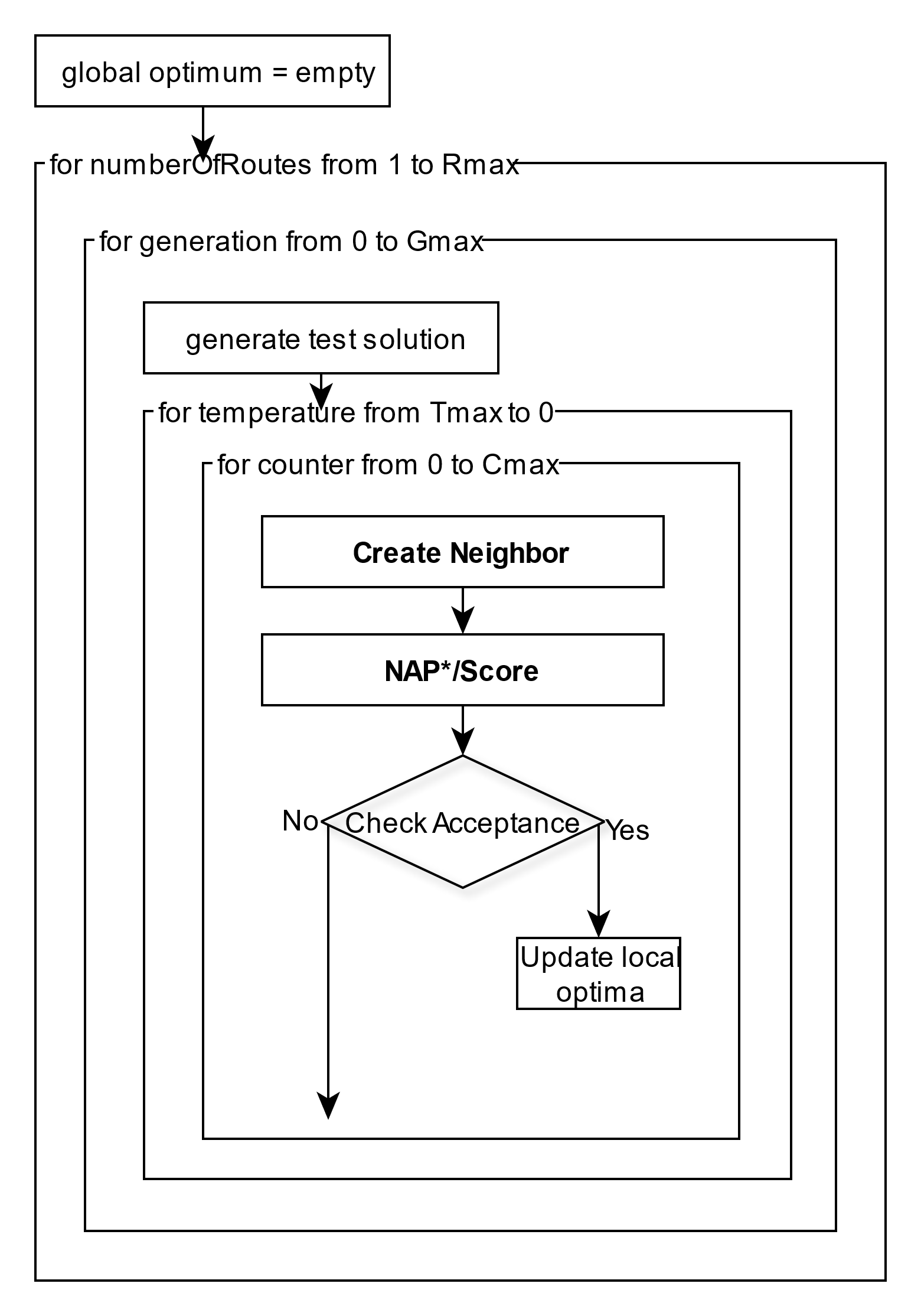
In order to generate continuous paths using the information included in the topology, the ICRSGP uses the Yen’s K-Shortest Path Algorithm [1]. Using an underlying shortest path algorithm called Dijkstra [1], this algorithm enables the generation of multiple optimal paths for a given pair of origin and destination. The setting parameter determines the maximum number of paths to created for each pair of nodes. As such K defines the runtime of this algorithm as well as the size of the resulting solution space.

In order to create K alternative paths that deviate from the initial shortest path between two nodes, this algorithm internally repeats the Dijkstra algorithm up to K times after isolating a different node of the initial path each time.

### Filtering for Constraints

After obtaining a set of potential paths by means of Yen’s K-Shortest Path Algorithm, a validation process will eliminate all paths that contradict the user’s demand regarding the length of routes, namely and . The ultimate size of the initial solution space will be determined by the balance of the setting parameter K and the restrictive property of the feasibility constraints.

## Metaheuristic Optimization Process



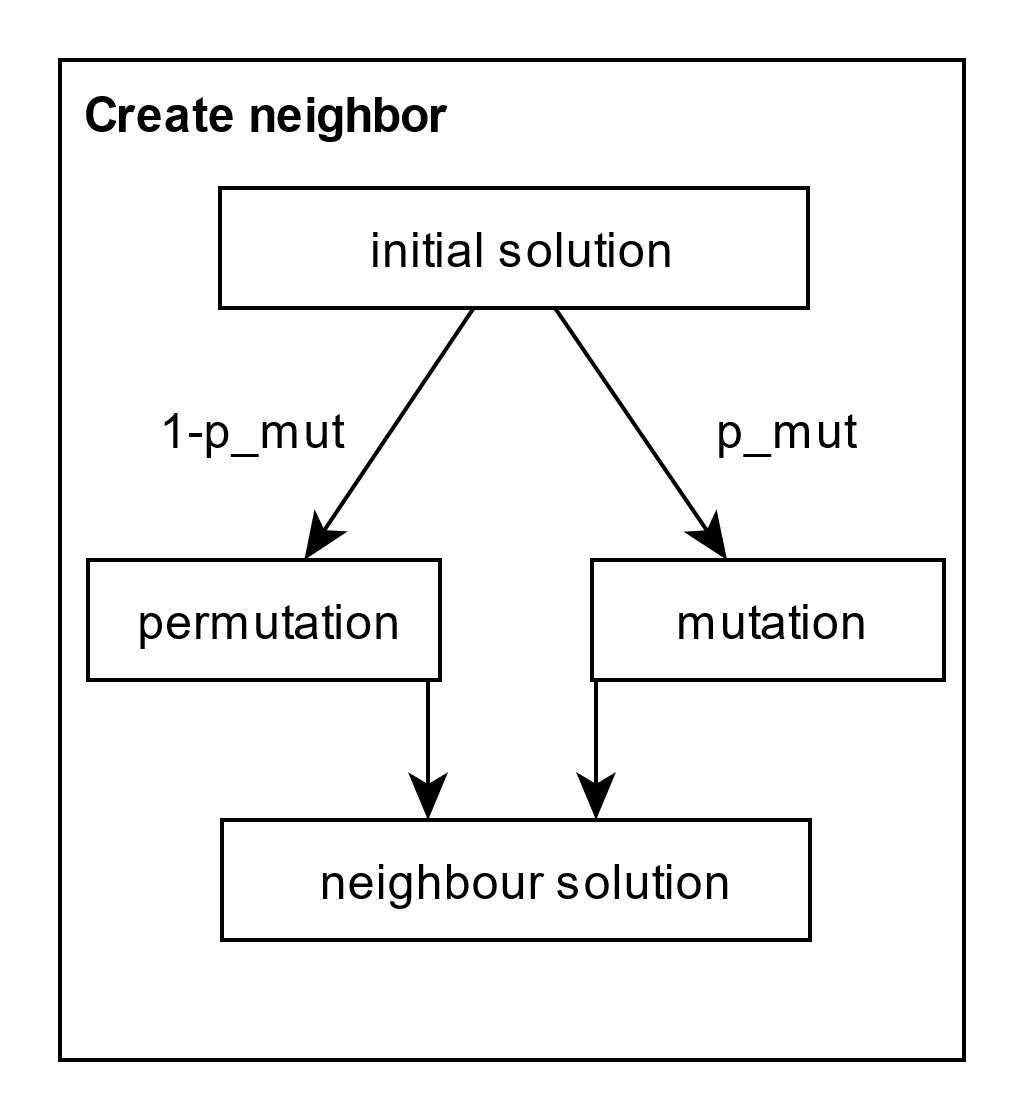
Using the before generated initial solution space the chosen algorithm will perform a metaheuristic search process as proposed in [1]. In Fig3 you can see a simplified overview of the algorithmic structure used in this approach.

### Traversing Solution Space

In order to traverse the solution in a directed manner, we will define a specific routine to generate solutions.

As proposed in [1] solutions are generated from ground up by the selection of a certain number of arbitrary paths from the initial solution space.

In addition to the generation of solution from ground up, there also need to be rules to derive neighboring derivation from an existing solution to a new one.



In contrast to the proposal in [1] we will create derivation not only through permutation, but also mutations of existing solutions. According the proposal neighboring solutions are to be generated by substituting single paths in the solution by other paths from the initial solution space. In reference to the practical experience with this algorithm we concluded the need to extend the initial solution space by allowing derivations using mutations of single paths within solutions. Paths can be mutated by extending and shrinking them by one node while considering existing feasibility constraints. Whether a neighbor is created by permutation of mutation will be decided randomly depending on p\_mut as the likelihood for the occurrence of a mutation.

### Simulated Annealing

The used approach is derived from the commonly known metaheuristic concept of “Simulated Annealing”. Inspired by thermodynamic processes Simulated Annealing can offer an efficient approach to complex optimization problems. [2]

While traversing the solution space this approach will keep mark of the last found optimal solution. Whenever a present solution improves upon the last known optimal solution, it will be accepted as the new optimum. Furthermore even if a present solution doesn’t lower the score it can be accepted with a certain probability determined by a steadily decreasing temperature value. The temperature profile will be defined by an exponential cooling function as explained in [1] with the factor as a setting parameter.

### NAP\*

As shown in Fig3 the approach includes a modified network analysis procedure (NAP\*) in comparison to the proposal in [1]. The NAP\* analyses any given test-solution using the user-defined constraints in order to calculate relevant measures. These measures encompass the frequencies of each route of the solution, as well as the score defined by the objective composition.

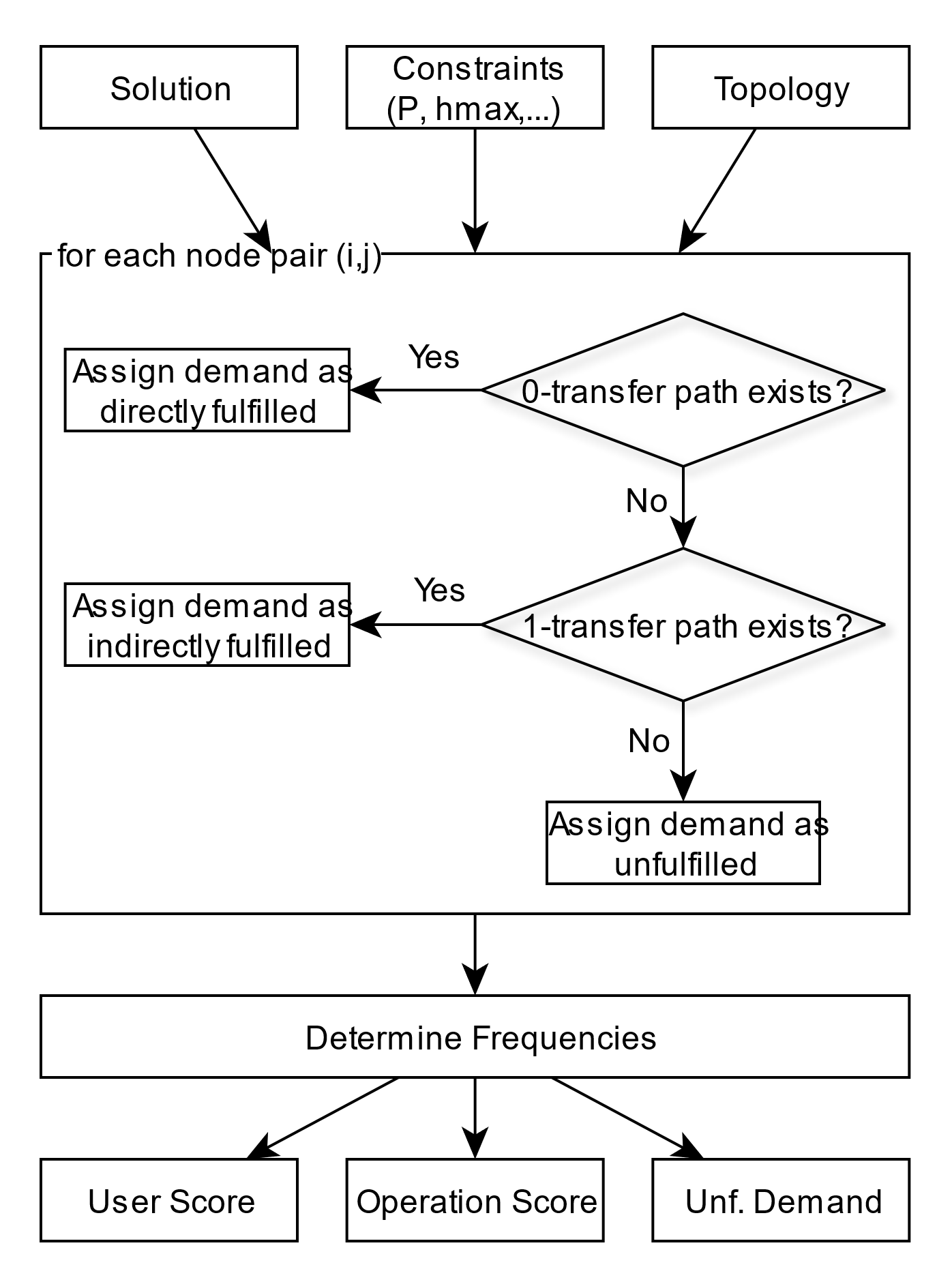


Fig4 illustrates the process of setting the respective frequencies to each route of a solution. For the purpose of frequency determination, the demand between each pair of nodes must be assigned to the respective routes. In reference to that we differentiate between demand that can be fulfilled using only one route (0-transfer), two routes (1-transfer) and categorize the rest as unfulfilled. In comparison to the proposal in […] we don’t accept 2-transfer paths. This modification was a result of considerations regarding the needed computation time. Calculating 2-transfer paths will quickly become a burden on the time consumption of the algorithm.

### Assign Load

In order to assign the demand into the correct categories, we will use the previously mentioned algorithm Dijkstra. After extracting a sub-topology that represents either a single route (0-transfer) or a combination of two routes (1-transfer), the Dijkstra algorithm will offer an insight into the feasibility of the demand as well as the corresponding parts of the routes that will be burdened by the demand.

Given the load-distribution over the network defined by the routes of the solution, we can determine the minimum frequencies needed to accommodate the demand by looking at the maximum load within each route. For a route I the definitions are introduced:

The following relation can be established:

### Score Calculation

The analysis of the solution will be finished by calculating its score in reference to the predefined objective composition of User Score, Operation Score and unfulfilled demand.

In the case of the User Score we will use the results of the categorization process and the travel time of the paths calculated by the Dijkstra algorithm.

User Score:

As the operation score describes the total number of vehicles running on the network defined by the current solution, we have to take the frequencies and the total travel time of each route into consideration.

Operation Score:

Finally our unfilled demand can be directly taken as the content of the respective category of the NAP\* algorithm used during the distribution of the load over the network.

### Setting Parameters

In the following section we will look at the newly introduced setting parameters as can be seen in Fig4.

|  |  |
| --- | --- |
| Parameter | Description |
| Gmax | Defines the number generation to be produced. Each generation is accompanied by the creation of a new initial solution from the ground inducing a new search path. |
| Tmax | Defines the number of steps within the steadily decreasing temperature profile/cooling function |
| Cmax | Defines the number of traversing steps taken for one starting solution and one temperature value. |

[1] Fan, Wei & B. Machemehl, Randy. (2006). Using a Simulated Annealing Algorithm to Solve the Transit Route Network Design Problem. Journal of Transportation Engineering-asce - J TRANSP ENG-ASCE. 132. 10.1061/(ASCE)0733-947X(2006)132:2(122).

[2] Carr, Roger. "Simulated Annealing." From MathWorld--A Wolfram Web Resource, created by Eric W. Weisstein. http://mathworld.wolfram.com/SimulatedAnnealing.html