Technical University of Denmark

DTU

Multi-Objective and Multi-Constrained Non-Additive Shortest Path Problems

Reinhardt, Line Blander; Pisinger, David

Link back to DTU Orbit

DTU Library
Technical Information Center of Denmark

Multi-Objective and Multi-Constrained Non-Additive Shortest Path Problems

Line Blander Reinhardt
David Pisinger
December 2009

Multi-Objective and Multi-Constrained Non-Additive Shortest Path Problems

Line Blander Reinhardt and David Pisinger[*]

December 8, 2009

Abstract

Shortest path problems appear as subproblems in numerous optimization problems. In most papers concerning multiple objective shortest path problems, additivity of the objective is a de-facto assumption, but in many real-life situations objectives and criteria, can be non-additive. The purpose of this paper is to give a general framework for dominance tests for problems involving a number of non-additive criteria. These dominance tests can help eliminate paths in a dynamic programming framework when using multiple objectives. Results on real-life multi-objective problems containing non-additive criteria are reported. We show that in many cases the framework can be used to efficiently reduce the number of generated paths.

Keywords: Multi objective programming, Shortest path problem, Non-additive objective, Dynamic programming

1 Introduction

The *shortest path problem* can be formulated on a directed graph $G = (V, E)$ where V is a finite set of vertices and E is a finite set of edges. The problem is to find a shortest path between a *source* $s \in V$ and a *destination* $t \in V$. In the *multi-objective shortest path problem* there are r criteria. An edge $e_{ij} \in E$ from vertex $i \in V$ to vertex $j \in V$ has associated values c_{ij}^k, $k \in \{1, ..., r\}$ for each criterion $k = 1, \ldots, r$. In order to have a well-defined problem it is assumed that there are no negative cycles for the criteria being minimized, or positive cycles for the criteria being maximized.

The general *additive* multi-objective shortest path problem with positive costs c_{ij} on the edges, and an additive objective function for each criterion, can be described by the following integer model (Martins [18]):

$$\text{minimize}: \quad z = \left(\sum_{(i,j) \in V} c_{ij}^1 e_{ij}, \quad \ldots \quad , \sum_{(i,j) \in V} c_{ij}^r e_{ij} \right) \tag{1}$$

$$s.t. \quad \sum_{j \in V} e_{sj} - \sum_{j \in V} e_{js} = 1 \tag{2}$$

$$\sum_{j \in V} e_{tj} - \sum_{j \in V} e_{jt} = -1 \tag{3}$$

$$\sum_{j \in V} e_{ij} - \sum_{j \in V} e_{ji} = 0 \quad \forall i \in V \setminus \{s, t\} \tag{4}$$

$$e_{ij} \in \{0, 1\} \quad \forall (i, j) \in E \tag{5}$$

---

[*]DTU Management, Technical University of Denmark, Produktionstorvet 424, DK-2800 Kgs.Lyngby, {lbre,pisinger}@man.dtu.dk

Constraint (2) states that there must be exactly one edge leaving s that is not part of a cycle. Constraint (3) states that there must be exactly one edge entering t that is not on a cycle. Constraint (4) is the ordinary flow conservation constraint. The solution z is an r vector which contains the values of the r objective functions for the path.

In real-life problems the objective functions may be *non-additive* and they may be functions of several criteria. In general the objective function has the following form:

$$\text{minimize} : z = \big(C^1(\mathcal{P}), \ldots, C^r(\mathcal{P}) \big) \tag{6}$$

where $\mathcal{P}$ is a path $\{s, \ldots, t\}$ and $C^1, \ldots, C^r$ are cost functions which for a given path $\mathcal{P}$ return a real number. We call this the *value vector*. A solution to the above problem is a set of all Pareto optimal paths. A path $\mathcal{P}$ is *Pareto optimal* if there is no other path $\mathcal{P}'$ between the same two vertices which is better or equal on all entries of the *value vector* and where at least one entry is better. The solution set where there is exactly one path for each Pareto optimal *value vector* is called the *minimal complete* set of Pareto optimal solutions.

Shortest path problems are among the most well-studied problems [1], however, results concerning multi-criteria problems are rare. This may be due to the fact that the monotonicity assumption of dynamic programming seldom holds for these problems. In [4] Carraway et al. describe monotonicity as the property of objectives preserving preferences for partial solutions in the dynamic programming recursion. This is not to be confused with non-decreasing or non-increasing functions. In shortest path problems the monotonicity criterion means that if $\mathcal{P}$ is a shortest path then the subpaths of $\mathcal{P}$ must also be shortest paths. For real life multi-criteria problems monotonicity only holds for special cases.

One of the reasons for the recent interest in multi-objective optimization is that optimization is being applied in public services and business applications. Therefore research in multi-objective shortest paths and attempts to circumvent the monotonicity assumption is of immediate interest.

Hansen [10] presented solution methods to monotone bicriteria and biobjective path problems using a label setting algorithm. Martins [18] generalized the label setting algorithm to an arbitrary number of objectives, however, monotonicity was still assumed. Brumbaugh-Smith and Shier [3] presented a label correcting algorithm to the multi-objective problem under the monotonicity assumption. As Ehrgott and Gandibleux mention [8], these problems were not extensively researched before the nineties. Several recent papers discuss approximation algorithms such as the FPTAS outlined by Tsaggouris and Zaroliagis [25]. For the exact solution methods Tsaggouris and Zaroliagis studied nonadditive paths with a single objective [24]. The objective is the sum of several criteria weighted by linear or non-linear coefficients. An early paper by Lengauer and Theune [17] mentions the problem of non-monotone cost structures with two criteria and shows that by changing the domination criteria the problems can be solved by using a standard shortest path algorithm such as Dijkstra's. Another paper considering a non-additive weighted objective is Carraway et al. [4] where the distance is minimized, and the probability of successfully reaching the destination is maximized. We will later show that both the probability and the distance criteria are monotone even though the sum of the two is not. It is shown in Carraway et al. [4] that it can be difficult to determine whether a multicriteria objective function is monotone. To circumvent the problem of not satisfying the monotonicity assumption Carraway et al. [4] introduced the concept of *generalized dynamic programming*, and presented a framework for solving multi-criteria shortest path problems. The framework, however, leaves it to the concrete application to define a local preference relation, that can be used to remove dominated states in the dynamic programming recursion. In this paper we present a number of such preference relations for the listed criteria functions. Moreover we give a general framework for how to define a local preference relation for non-additive objectives with certain general properties.

The contribution of this paper is to present a number of different criteria functions motivated by real-life applications and to develop an algorithm which finds all Pareto optimal solutions for a multi-objective shortest path problem. We present a general framework for dominance tests with all the presented criteria functions f and report computational experiments on a real-life instance with multiple criteria (addition, maximization, multiplication). Finding all Pareto-optimal solutions to a multi-objective shortest path problem has several similarities with the

solution of multi-constrained shortest-path problems [6, 9, 13, 14, 15, 21, 26]. The techniques developed in this paper can therefore be applied to several variants of multi-constrained shortest-path problems with non-additive constraints.

The paper differs from previous work as follows: Müller-Hannemann and Schnee [19] solve the nonadditive price problem by generally relaxing the Pareto optimality. The algorithm by Carraway et al. [4] only determines the *minimal complete* set of the Pareto optimal paths, and therefore does not find all Pareto optimal paths. Carraway uses a function u which maps all criteria to a real number and thereby only has one objective function. His framework does not specify how the dominance function should be implemented, but leaves it open to the concrete application. Irnich and Villeneuve [15] presented an algorithm for finding all Pareto-optimal solutions to a resource constrained shortest path problem with k-cycle eliminations. The considered (constrained and unconstrained) resources are all additive.

In the following Section 6.1, we present a number of multi-objective shortest path problems encountered in practice. Next, in Section 3, we formally define the set of Pareto optimal solutions and the corresponding dominance criterion. In Section 4, we present the fundamental dynamic programming algorithm used to solve the multi-objective shortest path problem for various cost functions. Section 5 provides a number of dominance rules for various cost functions that can be used to prune labels in a dynamic programming algorithm. Various monotone as well as non-monotone cost functions are considered. In Section 6, we return to the problems considered in Section 6.1 and discuss how the framework developed can be used to determine all Pareto-optimal solutions. Finally, Section 7 reports on computational experiments on real-life data from a shipping company. The paper concludes in Section 8.

2 The Multi-Objective Shortest Path Problems

Multi-criteria shortest path problems are well-studied for additive objective functions. However, in several real-life settings one cannot assume that the objective function is additive, neither can one assume that it is monotonously increasing. Let $w : E \to \mathbb{R}$ be an additive weight function on the edge weight and let f be a function from real numbers to real numbers, $f : \mathbb{R} \to \mathbb{R}$. The objective functions we will consider often contain the function $f \circ w$. The following list describes a number of objective functions which might be encountered in real life multi-criteria and multi-objective problems. Note that all of the objectives listed are non-additive:

A *Probability of reaching destination.* Carraway et al. [4] consider the objective of maximizing the probability of successfully reaching the destination. Each edge has an associated cost (length) and a probability for successful traversal. Assuming that probabilities are independent across edges, the probability of successfully reaching the destination is the product of the probabilities of the edges traversed. In Section 6.1, we will show how the product of probabilities can be converted to a function of the form $f \circ w$.

B *Combined distance and probability function.* Carraway et al.[4] consider an objective that is a combination of the distance d and the probability p. The objective is described as $-d + \lambda p, \quad \lambda \in \mathbb{R}_+$, where the aim is to maximize the objective and thereby minimize the length of the journey and maximize the probability. However, if the two criteria contradict each other the value of λ will affect how the two criteria are weighted. Referring to the objective of real life problem A, it is easy to see that this objective function will be of the form $-w_1 + f \circ w_2$ where $w_1, w_2 : E \to \mathbb{R}$ are additive.

C *Maximum of commissions.* Blander Reinhardt [2] describes a real life multi-objective cargo transportation problem in which each vertex corresponds to a hub port. Each time a vertex is visited (i.e. the cargo is reloaded) an agent is paid a commission. An agent may be responsible for several hubs, but will only be paid one commission. The commission paid will correspond to the largest commission the agent is entitled to on the path. The cost of a path is then the price of the edges plus the commission paid to agents. However,

only paying the agents the *largest* commission on the path complicates the objective. The objective function for the price objective is then $w + \sum_{a \in Agents} \max_a \{c_e^a | e \in E_P\}$, where E_P is the set of edges visited on the path $\mathcal{P}$ and c_e^a is the commission paid to agent a on edge e.

D *Number of zones visited.* Public transport in e.g. Copenhagen operates on a zone system [11]. Each zone covers a number of vertices (stations) and edges, and if a ticket is issued to a given zone, then it gives unlimited access to all vertices and edges in the zone. In other words our cost function implies that a cost is paid only the first time a zone is visited. A holder of a *monthly card* may buy access to any zone needed, hence the objective is to minimize the number of different zones on the path. Since a price is only paid the first time an edge in the given zone is visited, the objective function can be represented as $\sum_{z_h \in Zones} \max\{0, w_e | e \in E_{Z_h} \wedge e \in P\}$, where E_{Z_h} is the edges in the zone Z_h and $\mathcal{P}$ is the path travelled. The value w_e is 1 for all e.

E *Maximum zone distance from origin.* The cost of a *single-trip ticket* in public transport may depend on the maximum zone distance from the origin s, as in Copenhagen [11]. Here the vertices and the edges again belong to a zone. A one-zone ticket gives access to the zone containing s. A two-zone ticket gives access to all zones adjacent to the first zone. A three-zone ticket furthermore gives access to all zones adjacent to the previous zones. This means that if the non-starting zones visited on a trip all are neighboring to the starting zone s then a two zone ticket is needed even though more than two different zones maybe visited. The objective is to minimize the maximum zone distance between the zone of the edges on the path and the origin s to the destination t. In this case the objective function is as in D. However the zones are defined differently (see Section 6.5).

F *Zone distance and time.* In several public transportation ticket fare systems, travelers buy access to some zones in a given time period, hence it may be relevant to take both distance and time into consideration. A 1-zone ticket can be traversed within a time limit t_1, a 2-zone ticket can be traversed within a time limit $t_2 \geq t_1$, etc. The objective is to minimize travel cost. In this case the cost function takes a time and a number of zones and returns a cost. Here the objective function *obj* takes the maximum of a time function t and a zone function z, $obj = \max\{t, z\}$, where the zone function is as described in E and the time function t is of the form $f \circ w$.

G *Modulo k penalties.* Jepsen et al. [16] consider a shortest path problem where an additional penalty cost is paid for each of the k times nodes from a given set S have been visited. The objective is the cost of the edges plus a penalty depending on the number of times nodes from the set S have been visited. Here, the complicating factor is the penalty. This shortest path problem stems from the addition of Subset-Row inequalities [16] in a branch-and-price algorithm for the Vehicle Routing Problem. The objective function is $w_1 + f \circ w_2$, as in B, where f is the penalty function.

In the following sections we describe a general framework for solving multi-criteria shortest path problems. Then, in Section 5 we develop general schemes for non-additive functions of the form $f \circ w$. In Section 6, we return to the above problems and discuss how they can be solved using the schemes developed in Section 5.

3   Pareto Optimal Paths and Value Vectors

A path from s to t is denoted $\mathcal{P}_{st} = \{s, \ldots, t\}$. A sub-path $\mathcal{P}_{ij}$ of $\mathcal{P}_{st} = \{s, \ldots, i, \ldots, j \ldots, t\}$ is the path $\{i, \ldots, j\}$.

The optimal solution to the multi-objective shortest path problem (1) is a set of all Pareto optimal paths. A path is *Pareto optimal* if the value vector of that path is not *dominated* by the value vector of another path between the same two vertices. Let $x = (x_1, ..., x_r)$ and $y = (y_1, ..., y_r)$

be two real valued r vectors. For a minimization problem, x *dominates* y, if $x_k \leq y_k$ for all $k \in \{1, \ldots, r\}$ and $x_k < y_k$ for at least one $k \in \{1, \ldots, r\}$.

The *set of Pareto optimal paths* from a source s to a destination t is the set of paths from s to t with non-*dominated* value vectors. It should be noted that there can be several Pareto optimal paths with the same value vector.

If the problem has more than one objective then there can be an exponential number of Pareto optimal paths where each has a unique value vector. In Hansen [10] a graph is presented where there is an exponential number of paths from x_1 to x_n which are all Pareto optimal and have unique value vectors. However, as observed by Müller-Hannemann and Weihe [20], the number of Pareto optimal paths in real-life problems is usually quite small. This observation is also confirmed in the real-life problems we have studied.

In some problems only the set of *minimal complete* Pareto optimal paths are sought. The *minimal complete* set of Pareto optimal paths was defined in [10] for bicriteria problems. The *minimal complete* Pareto optimal paths are, as defined in Section 1, the set containing exactly one path per Pareto optimal value vector. Note that the *minimal complete* set of Pareto optimal paths also can contain exponentially many paths [10]. When a shortest path problem contains only two objective functions the *minimal complete* set of Pareto optimal paths can be found using an integer programming method called the Ranking method. At each iteration of the ranking method a Pareto optimal value vector and a path satisfying the value vector is found (see [8] for further details). However, we have no knowledge of a constructive way of finding the *minimal complete* set of Pareto optimal paths for problems with more than two objective functions.

It should be mentioned that in quite a few real life problems it would be desirable to find not only the *minimal complete* set of Pareto optimal paths but *all* the Pareto optimal paths. This is in particular true in cases where there is a decision maker who selects the most desirable solution. It is reasonable to assume that there are factors unknown to the program that depend entirely on the specific decision maker. Therefore, two different paths with the same objective values might be viewed differently by the decision maker.

4 Dynamic Programming

We will use dynamic programming to find the set of pareto optimal paths. Dynamic programming relies on the *principle of optimality*. For multi-objective problems where one or more objectives does not satisfy the monotonicity property, one must use the *weak principle of optimality* as defined by Carraway et al. [4]

Principle of optimality An optimal path must be composed of optimal subpaths.

Weak principle of optimality An optimal path must be composed of subpaths that can be part of an optimal path.

Irnich and Villeneuve [15] defined a similar weak principle of optimality for multi-constrained shortest path problems based on the concept of *extensions* $\mathcal{E}(\mathcal{P})$ of a given subpath $\mathcal{P}$.

There are two general dynamic programming algorithms for additive multi-objective shortest path problems based on Dijkstra's algorithm for the single objective shortest path problem [5]. These algorithms are the *Label-Setting* algorithm [18] and the *Label-Correcting* algorithm [22]. The Label-Setting algorithm does not allow negative edge costs. The Label-Correcting algorithm does allow negative edge costs but no negative cycles. We will apply the Label-Correcting algorithm in the sequel.

In the pseudocode for this algorithm let $C^1(\mathcal{P}) \ldots C^r(\mathcal{P})$ be the cost function of a path $\mathcal{P}$ and let `Merge` be a function, which, given two sets of labels, returns only the undominated labels of the union of the two sets. The set Q consists of the vertices with undominated labels that have not yet been used to generate other labels. Each label is given as the tuple $(C^1(\mathcal{P}), \ldots, C^r(\mathcal{P}), \text{pred}(\mathcal{P}))$, where $\text{pred}(\mathcal{P})$ is a pointer to the label it was generated from. The label correcting algorithm is outlined in the following pseudocode, inspired by [3] and [22]. In each node v we maintain

a list L_v of labels. Clearly, by following the pred($\mathcal{P}$) pointers backward from a vertex v to the source s one gets the subpath the label represents. Thus, each label in the lists L_v represents a subpath $\mathcal{P} = \{s, \ldots, v\}$ which is not dominated by other subpaths from s to v. The set Q can with advantage be implemented as an lexicographically ordered list.

LABEL-CORRECTING(G, s, t)

1: $L_v \leftarrow \emptyset$ **for all** $v \in V \setminus \{s\}$
2: $L_s \leftarrow \{(0, \ldots, 0, \{s\}))\}$;
3: $Q \leftarrow \{s\}$;
4: **while** $Q \neq \emptyset$ **do**
5:    $u \leftarrow$ **extract vertex from** Q;
6:    **for all edges** e_{uv} **do**
7:       $L'_v \leftarrow \texttt{Merge}(L_v, L_u \circ \{e_{uv}\})$;
8:       **if** $L'_v \neq L_v$ **then**
9:          $L_v \leftarrow L'_v$;
10:          $Q \leftarrow Q \cup \{v\}$
11:       **end if**
12:    **end for**
13: **end while**
14: **return** L_v **for all** $v \in V$;

The label correcting algorithm repeatedly extracts a vertex u from the set Q, and for each outgoing edge e_{uv} extends the labels $(C^1(\mathcal{P}_{su}), \ldots, C^r(\mathcal{P}_{su}), \text{pred}(\mathcal{P}_{su}))$ in L_u to $(C^1(\mathcal{P}_{sv}), \ldots, C^r(\mathcal{P}_{sv}), \text{pred}(\mathcal{P}_{sv}))$. These new labels $L_u \cup \{e_{uv}\}$ are then $\texttt{Merge}$'d together with the old labels L_v of v. The merging eliminates *dominated* labels. If the set of labels at v has been changed during the $\texttt{Merge}$, then v is added to Q. This is repeated until the set Q is empty. The $\circ$ operator in line 7 must match the objective functions, and the dominance criterion used implicit in $\texttt{Merge}$ must be tailored to return the undominated labels. We will in the next section propose various sufficient criteria for removing dominated labels.

5 Non-Additive Objectives in Dynamic Programming

As mentioned before, dynamic programming algorithms for shortest path problems rely on the monotonicity requirement. The monotonicity requirement ensures that subpaths of a Pareto optimal path are Pareto optimal and therefore the subpaths that are not Pareto optimal can be eliminated from the search. Clearly the additive case is monotone, however there exists other monotone objective functions. Theorem 1 covers a set of objective functions which satisfies the monotonicity requirement. It should be noted that Theorem 1 is not exhaustive, as other objectives may exist that satisfy the monotonicity requirement.

Section 5.1 defines a dominance criterion for problems where the objective function is based on two or more additive weight functions. Section 5.2 defines dominance criteria for problems where the objective function is defined as the maximum of a set of edge weights visited on the path. The recently published example of optimizing the mean and variance of a random variable associated with an edge described in [12] by Hutson and Shier shows a non-additive objective of the structure described in Section 5.1. Weight functions evaluate a single criterion along a path. However an objective can include several criteria. Thus several weight functions including the weight of a criterion on a path can be transformed by one function in the objective. An additive objective requires that the weight functions (also called criteria) included in it are additive, however, an objective on one or more additive weight functions is not necessarily additive as the additive weight function can be part of a non-additive function in the objective.

Theorem 1 *Given a weighted directed graph $G = (V, E)$ with an additive weight function $w : E \to \mathbb{R}^k$ ($w_\ell : E \to \mathbb{R}$, $\ell \in \{1, .., k\}$) where k is the number of objective functions, let the objective*

functions be $C_\ell = f_\ell \circ w_\ell$ where $f_\ell : \mathbb{R} \to \mathbb{R}$ is a strictly increasing or strictly decreasing function. Let $\mathcal{P}_{st}$ be a Pareto optimal path from s to t, then any subpath $\mathcal{P}_{ij}$ of $\mathcal{P}_{st}$ is a Pareto optimal path from i to j.

The theorem shows that if the objective functions are strictly increasing or strictly decreasing, then the normal *principle of optimality* holds, and hence we can use an ordinary dominance rule in the label correcting algorithm.

Proof First assume that the objective functions are to be minimized. Assume that $\mathcal{P}_{ij}$ is not a Pareto minimal path, then there would be a path $\mathcal{P}'_{ij}$ that dominates the path $\mathcal{P}_{ij}$. Decompose the Pareto minimal path $\mathcal{P}_{st}$ into three subpaths $\mathcal{P}_{si}$, $\mathcal{P}_{ij}$ and $\mathcal{P}_{jt}$. Then, because of the additive structure of the weight function, we have $w(\mathcal{P}_{st}) = w(\mathcal{P}_{si}) + w(\mathcal{P}_{ij}) + w(\mathcal{P}_{jt})$. Let the path $\mathcal{P}'_{st}$ be defined by subpaths $\mathcal{P}_{si}$, $\mathcal{P}'_{ij}$ and $\mathcal{P}_{jt}$. Clearly the weight of $\mathcal{P}'_{st}$ is $w(\mathcal{P}'_{st}) = w(\mathcal{P}_{si}) + w(\mathcal{P}'_{ij}) + w(\mathcal{P}_{jt})$. Since $\mathcal{P}'_{ij}$ dominates $\mathcal{P}_{ij}$ we have $f_\ell(w_\ell(\mathcal{P}'_{ij})) \le f_\ell(w_\ell(\mathcal{P}_{ij}))$ where the inequality is strict for at least one $\ell \in \{1, ..., k\}$. Then, in the case where f_ℓ is strictly *increasing*, one gets the following inequalities where for at least one ℓ the inequality is strict:

$$f_\ell(w_\ell(\mathcal{P}'_{ij})) \le f_\ell(w_\ell(\mathcal{P}_{ij})) \;\Rightarrow\; w_\ell(\mathcal{P}'_{ij}) \le w_\ell(\mathcal{P}_{ij}) \;\Rightarrow$$
$$w_\ell(\mathcal{P}'_{st}) \le w_\ell(\mathcal{P}_{st}) \;\Rightarrow\; f_\ell(w_\ell(\mathcal{P}'_{st})) \le f_\ell(w_\ell(\mathcal{P}_{st}))$$

In the case where f_ℓ is strictly *decreasing* one gets a similar result by reversing the appropriate inequalities. Therefore the path $\mathcal{P}'_{st}$ dominates the path $\mathcal{P}_{st}$, which contradicts the Pareto minimality of the path $\mathcal{P}_{st}$. For maximization the proof is similar.

$\square$

Note that Theorem 1 also holds for graphs with negative weights and that an additive objective function with nonnegative edge costs c_{ij}^k is a special case of the strictly increasing function.

5.1 Objectives Based on Additive Weight Functions

We will start by showing the case of an objective based on a finite number of additive weight functions. The problem for non-additive objectives is that the value of $C(\mathcal{P}_{it})$ can vary depending on the path taken from s to i. This is also the case when the non-additive objective is based on additive weight functions.

Only the single objective case is considered, although the results easily can be generalized to the multi-objective case by using the definition of Pareto optimality.

Theorem 2 (Gradient domination) *Given a weighted directed graph $G = (V, E)$ let $w_i : E \to \mathbb{R}, i \in \{1, \ldots, n\}$ be additive weight functions on G. Let there be an objective function of the form: $C(\mathcal{P}) = \sum_{i=1}^n f_i(w_i(\mathcal{P}))$. Let $\mathcal{P}_{st}$ be composed of subpaths $\mathcal{P}_{sj}$ and $\mathcal{P}_{jt}$ and let $\mathcal{P}'_{st}$ be composed of $\mathcal{P}'_{sj}$ and $\mathcal{P}_{jt}$ (see Figure 1). Let*

$$\mathcal{M}_i^- \le \frac{f_i(w_i(\mathcal{P}'_{st})) - f_i(w_i(\mathcal{P}_{st}))}{w_i(\mathcal{P}'_{st}) - w_i(\mathcal{P}_{st})} \le \mathcal{M}_i^+, \quad i \in \{1, \ldots, n\}, w_i(\mathcal{P}_{st}) \ne w_i(\mathcal{P}'_{st}) \tag{7}$$
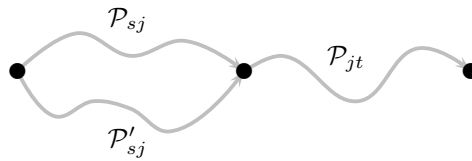

Figure 1: Path from s to t is split into $\mathcal{P}_{sj}, \mathcal{P}'_{sj}, \mathcal{P}_{jt}$.

where $\mathcal{M}_i^-, \mathcal{M}_i^+ \in \mathbb{R}$. Moreover let:

$$\sum_{i=1}^{n} \mathcal{M}_i w_i(\mathcal{P}'_{sj}) < \sum_{i=1}^{n} \mathcal{M}_i w_i(\mathcal{P}_{sj}) \tag{8}$$

with strict inequality for at least one i, where

$$\mathcal{M}_i = \begin{cases} \mathcal{M}_i^+ & \text{if } w_i(\mathcal{P}'_{sj}) > w_i(\mathcal{P}_{sj}) \\ 1 & \text{if } w_i(\mathcal{P}'_{sj}) = w_i(\mathcal{P}_{sj}) \\ \mathcal{M}_i^- & \text{if } w_i(\mathcal{P}'_{sj}) < w_i(\mathcal{P}_{sj}) \end{cases}$$

Then $C(\mathcal{P}'_{st}) < C(\mathcal{P}_{st})$.

Proof Assume that (8) holds. For each i where $w_i(\mathcal{P}_{sj}) \neq w_i(\mathcal{P}'_{sj})$ we have:

$$\frac{f_i(w_i(\mathcal{P}'_{st})) - f_i(w_i(\mathcal{P}_{st}))}{w_i(\mathcal{P}'_{sj}) - w_i(\mathcal{P}_{sj})} = \frac{f_i(w_i(\mathcal{P}'_{st})) - f_i(w_i(\mathcal{P}_{st}))}{w_i(\mathcal{P}_{jt}) + w_i(\mathcal{P}'_{sj}) - (w_i(\mathcal{P}_{sj}) + w_i(\mathcal{P}_{jt}))} = \frac{f_i(w_i(\mathcal{P}'_{st})) - f_i(w_i(\mathcal{P}_{st}))}{w_i(\mathcal{P}'_{st}) - w_i(\mathcal{P}_{st})}$$

which is less than or equal to $\mathcal{M}_i^+$ if $w_i(\mathcal{P}'_{sj}) > w_i(\mathcal{P}_{sj})$ and which is greater than or equal to $\mathcal{M}_i^-$ if $w_i(\mathcal{P}'_{sj}) < w_i(\mathcal{P}_{sj})$. This implies that

$$f_i(w_i(\mathcal{P}'_{st})) - f_i(w_i(\mathcal{P}_{st})) \leq \mathcal{M}_i(w_i(\mathcal{P}'_{st}) - w_i(\mathcal{P}_{st})) \tag{9}$$

It is easy to see that inequality (9) also will hold if $w_i(\mathcal{P}_{sj}) = w_i(\mathcal{P}'_{sj})$. This means that

$$\sum_{i=1}^{n} (f_i(w_i(\mathcal{P}'_{st})) - f_i(w_i(\mathcal{P}_{st}))) \leq \sum_{i=1}^{n} (\mathcal{M}_i(w_i(\mathcal{P}'_{st}) - w_i(\mathcal{P}_{st})))$$

and hence

$$\sum_{i=1}^{n} (f_i(w_i(\mathcal{P}'_{st})) \leq \sum_{i=1}^{n} (\mathcal{M}_i(w_i(\mathcal{P}'_{st}) - w_i(\mathcal{P}_{st}))) + \sum_{i=1}^{n} f_i(w_i(\mathcal{P}_{st})) \tag{10}$$

Now, adding inequalities (8) and (10) we get:

$$\sum_{i=1}^{n} (f_i(w_i(\mathcal{P}'_{st})) < \sum_{i=1}^{n} (f_i(w_i(\mathcal{P}_{st}))$$

and hence $C(\mathcal{P}'_{st}) < C(\mathcal{P}_{st})$ which proves the theorem.

$\square$

In the next corollary we consider problems with additive weight functions where an objective function is based on two additive criteria. This case is relevant for some of the real life problems considered in Section 6.

Corollary 1 (minimization of objective) *Given a weighted directed graph $G = (V, E)$ let $w_1 : E \rightarrow \mathbb{R}$ and $w_2 : E \rightarrow \mathbb{R}$ be two additive weight functions on G. Let there be an objective function of the form: $C(\mathcal{P}) = w_1(\mathcal{P}) + f(w_2(\mathcal{P}))$ which is to be minimized. Let $\mathcal{P}_{st}$ be composed of subpaths $\mathcal{P}_{sj}$ and $\mathcal{P}_{jt}$ and let $\mathcal{P}'_{st}$ be composed of $\mathcal{P}'_{sj}$ and $\mathcal{P}_{jt}$ (see Figure 1). Let*

$$\mathcal{M}^- \leq \frac{f(w_2(\mathcal{P}'_{st})) - f(w_2(\mathcal{P}_{st}))}{w_2(\mathcal{P}'_{st}) - w_2(\mathcal{P}_{st})} \leq \mathcal{M}^+, \ w_2(\mathcal{P}'_{st}) \neq w_2(\mathcal{P}_{st}) \tag{11}$$

where $\mathcal{M}^-, \mathcal{M}^+ \in \mathbb{R}$. Moreover let:

$$\begin{array}{ll} w_1(\mathcal{P}'_{sj}) + \mathcal{M}^+ w_2(\mathcal{P}'_{sj}) < w_1(\mathcal{P}_{sj}) + \mathcal{M}^+ w_2(\mathcal{P}_{sj}) & \text{if } w_2(\mathcal{P}'_{sj}) > w_2(\mathcal{P}_{sj}) \\ w_1(\mathcal{P}'_{sj}) + \mathcal{M}^- w_2(\mathcal{P}'_{sj}) < w_1(\mathcal{P}_{sj}) + \mathcal{M}^- w_2(\mathcal{P}_{sj}) & \text{if } w_2(\mathcal{P}'_{sj}) < w_2(\mathcal{P}_{sj}) \\ w_1(\mathcal{P}'_{sj}) < w_1(\mathcal{P}_{sj}) & \text{if } w_2(\mathcal{P}'_{sj}) = w_2(\mathcal{P}_{sj}) \end{array} \tag{12}$$

Then $C(\mathcal{P}'_{st}) < C(\mathcal{P}_{st})$.

Corollary 1 states that if two subpaths $\mathcal{P}_{sj}$ and $\mathcal{P}'_{sj}$ ending at the same node j satisfy inequality (12), then for minimization problems the path $\mathcal{P}'_{sj}$ dominates $\mathcal{P}_{sj}$ and the latter may be deleted.

Proof This is the special case of Theorem 2 where $n = 2$ and f_1 is the identity function.

$\square$

The dominance rule (12) was defined for a minimization problem. In the case of *maximization* Corollary 1 is changed to

Corollary 2 (maximization of objective) *Given a weighted directed graph $G = (V, E)$ let $w_1 : E \to \mathbb{R}$ and $w_2 : E \to \mathbb{R}$ be two additive weight functions on G. Let there be an objective function of the form: $C(p) = w_1(p) + f(w_2(p))$ which is to be maximized. Let $\mathcal{P}_{st}$ be composed of subpaths $\mathcal{P}_{sj}$ and $\mathcal{P}_{jt}$ and let $\mathcal{P}'_{st}$ be composed of $\mathcal{P}'_{sj}$ and $\mathcal{P}_{jt}$ (See Figure 1) . Let*

$$\mathcal{M}^- \leq \frac{f(w_2(\mathcal{P}'_{st})) - f(w_2(\mathcal{P}_{st}))}{w_2(\mathcal{P}'_{st}) - w_2(\mathcal{P}_{st})} \leq \mathcal{M}^+, \; w_2(\mathcal{P}'_{st}) \neq w_2(\mathcal{P}_{st}) \tag{13}$$

where $\mathcal{M}^-, \mathcal{M}^+ \in \mathbb{R}$. Moreover let:

$$\begin{array}{lll} w_1(\mathcal{P}'_{sj}) + \mathcal{M}^+ w_2(\mathcal{P}'_{sj}) > w_1(\mathcal{P}_{sj}) + \mathcal{M}^+ w_2(\mathcal{P}_{sj}) & if & w_2(\mathcal{P}'_{sj}) < w_2(\mathcal{P}_{sj}) \\ w_1(\mathcal{P}'_{sj}) + \mathcal{M}^- w_2(\mathcal{P}'_{sj}) > w_1(\mathcal{P}_{sj}) + \mathcal{M}^- w_2(\mathcal{P}_{sj}) & if & w_2(\mathcal{P}'_{sj}) > w_2(\mathcal{P}_{sj}) \\ w_1(\mathcal{P}'_{sj}) > w_1(\mathcal{P}_{sj}) & & if & w_2(\mathcal{P}'_{sj}) = w_2(\mathcal{P}_{sj}) \end{array} \tag{14}$$

Then $C(\mathcal{P}'_{st}) > C(\mathcal{P}_{st})$.

In finding $\mathcal{M}^-$ and $\mathcal{M}^+$ the goal is to maximize $\mathcal{M}^-$ and minimize $\mathcal{M}^+$ so that the number of paths kept for investigation is minimized.

Remark 1 When f is **differentiable** and there exists $\mathcal{M}^-$ and $\mathcal{M}^+$ such that $\mathcal{M}^- \leq f'(x) \leq \mathcal{M}^+$ for all x in the domain of f, then by the Mean Value Theorem we have

$$\mathcal{M}^- \leq \frac{f(w_2(\mathcal{P}'_{st})) - f(w_2(\mathcal{P}_{st}))}{w_2(\mathcal{P}'_{st}) - w_2(\mathcal{P}_{st})} \leq \mathcal{M}^+.$$

Note that on a fixed weight graph G the domain of f can be restricted to a closed interval $[a, b]$ such that $f(w_2(P)) \in [a, b]$ for every simple path P in G. Moreover there could be a lower bound on how much $w_2(\mathcal{P}'_{st}) - w_2(\mathcal{P}_{st})$ can be for two different paths. This could for example be the smallest cost of an edge in G.

Such upper bound on $f(w_2(\mathcal{P}'_{st})) - f(w_2(\mathcal{P}_{st}))$ and lower bound $w_2(\mathcal{P}'_{st}) - w_2(\mathcal{P}_{st})$ could be used to find a possible value for $\mathcal{M}^+$. In the same way possible values for $\mathcal{M}^-$ can be found. Clearly if f is a differentiable bounded function then $\mathcal{M}^+$ can be the maximum of the derivative and $\mathcal{M}^-$ the minimum of the derivative in the bounded region.

Remark 2 When f is **convex** and $w_2 : E \to \mathbb{R}_0^+$ then

$$\frac{f(w_2(\mathcal{P}_{si})) - f(w_2(\mathcal{P}'_{si}))}{w_2(\mathcal{P}_{si}) - w_2(\mathcal{P}'_{si})} \leq \frac{f(w_2(\mathcal{P}_{st})) - f(w_2(\mathcal{P}'_{st}))}{w_2(\mathcal{P}_{st}) - w_2(\mathcal{P}'_{st})} = \frac{f(w_2(\mathcal{P}'_{st})) - f(w_2(\mathcal{P}_{st}))}{w_2(\mathcal{P}'_{st}) - w_2(\mathcal{P}_{st})} \tag{15}$$

and thus $\mathcal{M}^-$ can be chosen as:

$$\mathcal{M}^- = \frac{f(w_2(\mathcal{P}_{si})) - f(w_2(\mathcal{P}'_{si}))}{w_2(\mathcal{P}_{si}) - w_2(\mathcal{P}'_{si})} \tag{16}$$

when substituting the value of $\mathcal{M}^-$ into the portion of the dominance definition in Corollary 1 containing $\mathcal{M}^-$ (both in the minimization and maximization versions) then regular dominance (principle of optimality) is achieved in that portion.
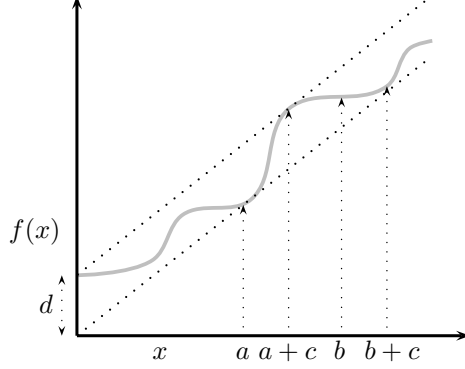
9

Figure 2: The property of f to facilitate the use of Floor domination

Moreover if $w_2(\mathcal{P}_{it}) \leq b$ for all simple paths $\mathcal{P}_{it}$ from i to t in G then we can choose $\mathcal{M}^+$ as follows:

$$\mathcal{M}^+ = \frac{f(b + w_2(\mathcal{P}'_{si})) - f(b + w_2(\mathcal{P}_{si}))}{w_2(\mathcal{P}'_{si}) - w_2(\mathcal{P}_{si})} \geq \frac{f(w_2(\mathcal{P}'_{st})) - f(w_2(\mathcal{P}_{st}))}{w_2(\mathcal{P}'_{st}) - w_2(\mathcal{P}_{st})}$$

Similarly when $w_2 : E \to \mathbb{R}_0^-$ then $\mathcal{M}^+$ can be chosen as

$$\mathcal{M}^+ = \frac{f(w_2(\mathcal{P}'_{si})) - f(w_2(\mathcal{P}_{si}))}{w_2(\mathcal{P}'_{si}) - w_2(\mathcal{P}_{si})} \geq \frac{f(w_2(\mathcal{P}'_{st})) - f(w_2(\mathcal{P}_{st}))}{w_2(\mathcal{P}'_{st}) - w_2(\mathcal{P}_{st})}, \tag{17}$$

which in the portion containing $\mathcal{M}^+$ reduces to regular dominance in the dominance definition of Corollary 1. If $\exists a \leq w_2(\mathcal{P}_{it})$ for all simple paths $\mathcal{P}_{it}$ from i to t in G then $\mathcal{M}^-$ can be chosen as

$$\mathcal{M}^- = \frac{f(a + w_2(\mathcal{P}_{si})) - f(a + w_2(\mathcal{P}'_{si}))}{w_2(\mathcal{P}_{si}) - w_2(\mathcal{P}'_{si})} \leq \frac{f(w_2(\mathcal{P}_{st})) - f(w_2(\mathcal{P}'_{st}))}{w_2(\mathcal{P}_{st}) - w_2(\mathcal{P}'_{st})}. \tag{18}$$

Remark 3 When f is **concave**, $-f$ is convex and therefore we can apply Remark 2 to $-f$ to find suitable values of $\mathcal{M}^-$ and $\mathcal{M}^+$ for f.

These remarks will be used when returning to the objective functions described in Section 6.1.

Another function used in the objectives described in Section 6 is the *floor* function. Clearly the *floor* function fulfills the requirements of Corollary 1, however, $\mathcal{M}^+$ is infinite. Using the *floor* dominance stated by Jepsen et al. in [16], we present a general form of domination.

Theorem 3 (Floor domination) *Given a weighted directed graph $G = (V, E)$ let $w_1 : E \to \mathbb{R}$ and $w_2 : E \to \mathbb{R}$ be two additive weight functions on G. Let there be an objective function of the form: $C(\mathcal{P}) = w_1(\mathcal{P}) + f(w_2(\mathcal{P}))$. Let the function $f: \mathbb{R} \to \mathbb{R}$ satisfy $f(a + c) - f(a) - d \leq f(b + c) - f(b)$ for all a,b,c in the domain of f (see Figure 2) where a, b satisfy some property D and $d \in \mathbb{R}_0^+$. Let $\mathcal{P}_{st}$ be composed of subpaths $\mathcal{P}_{sj}$ and $\mathcal{P}_{jt}$ and let $\mathcal{P}'_{st}$ be composed of $\mathcal{P}'_{sj}$ and $\mathcal{P}_{jt}$ (see Figure 1). Then:*

$$C(\mathcal{P}'_{sj}) + d \leq C(\mathcal{P}_{sj}) \Rightarrow C(\mathcal{P}'_{st}) \leq C(\mathcal{P}_{st}) \tag{19}$$

when $w_2(\mathcal{P}'_{sj}), w_2(\mathcal{P}_{sj})$ are satisfying property D.

Proof Assume $w_1(\mathcal{P}'_{sj}) + f(w_2(\mathcal{P}'_{sj})) + d \leq w_1(\mathcal{P}_{sj}) + f(w_2(\mathcal{P}_{sj}))$ and that $w_2(\mathcal{P}'_{sj}), w_2(\mathcal{P}_{sj})$ satisfy a property D. Then, by the additivity of w_1 we have that:

$$w_1(\mathcal{P}'_{st}) + f(w_2(\mathcal{P}'_{sj})) + d \leq w_1(\mathcal{P}_{st}) + f(w_2(\mathcal{P}_{sj})) \tag{20}$$

By the property of f we have:

$$f(w_2(\mathcal{P}'_{sj}) + w_2(\mathcal{P}_{jt})) - f(w_2(\mathcal{P}'_{sj})) - d \leq f(w_2(\mathcal{P}_{sj}) + w_2(\mathcal{P}_{jt})) - f(w_2(\mathcal{P}_{sj})) \tag{21}$$

10

by adding the previous two inequalities we get:

$$w_1(\mathcal{P}'_{st}) + f(w_2(\mathcal{P}'_{st})) \leq w_1(\mathcal{P}_{st}) + f(w_2(\mathcal{P}_{st})) \tag{22}$$

$\square$

Theorem 3 is applied to real life problems in Sections 6.6 and 6.7, and examples of the determination of d and property D are presented.

5.2 Objectives Based on the Max and Min Function

In this subsection we consider a non-monotone objective function defined as the maximum of a set of edge weights visited on the path $\mathcal{P}$. Theorem 4 and its proof is a generalization of the work in [2].

Theorem 4 (Max domination) *Given a weighted directed graph $G = (V, E)$ let $w_1 : E \to \mathbb{R}$ be an additive weight function and $a_j : E \to \mathbb{R}$, $j \in \{1, ..., n\}$ be a map from the edges of G into $\mathbb{R}$ and let $e \in E$ be an edge in G. Then, let $w_2(\mathcal{P}) = \sum_{j=1}^{n} \max_{e \in \mathcal{P}} a_j(e)$. Let there be an objective function of the form $C(p) = w_1(p) + w_2(p)$ to be minimized. Let $\mathcal{P}_{st}$ be composed of subpaths $\mathcal{P}_{si}$ and $\mathcal{P}_{it}$ and let $\mathcal{P}'_{st}$ be composed of $\mathcal{P}'_{si}$ and $\mathcal{P}_{it}$. Moreover let:*

$$w_1(\mathcal{P}'_{si}) + \sum_{j=1}^{n} \max_{e \in \mathcal{P}'_{si}} \{a_j(e)\} < w_1(\mathcal{P}_{si}) + \sum_{j=1}^{n} \max_{e \in \mathcal{P}_{si}} \{a_j(e)\} - F(\mathcal{P}_{si}, \mathcal{P}'_{si}) \tag{23}$$

where

$$F(\mathcal{P}, \mathcal{P}') = \sum_{j=1}^{n} \max\{0, \max_{e \in \mathcal{P}} a_j(e) - \max_{e \in \mathcal{P}'} a_j(e)\}.$$

Then $C(\mathcal{P}'_{st}) < C(\mathcal{P}_{st})$.

In other words if two subpaths $\mathcal{P}_{si}$ and $\mathcal{P}'_{si}$ end at the same node i and (23) is satisfied, then the label corresponding to subpath $\mathcal{P}'_{si}$ dominates the label corresponding to subpath $\mathcal{P}_{si}$ and hence the latter may be deleted.

Proof With some trivial case studies it is easy to see that for each $j = 1, \ldots, n$ we have

$$\max\left\{0, \max_{e \in \mathcal{P}_{it}}\{a_j(e)\} - \max_{e \in \mathcal{P}'_{si}}\{a_j(e)\}\right\} - \max\left\{0, \max_{e \in \mathcal{P}_{it}}\{a_j(e)\} - \max_{e \in \mathcal{P}_{si}}\{a_j(e)\}\right\}$$

$$\leq \max\left\{0, \max_{e \in \mathcal{P}_{si}}\{a_j(e)\} - \max_{e \in \mathcal{P}'_{si}}\{a_j(e)\}\right\}$$

Adding these inequalities for $j = 1, \ldots, n$, we obtain

$$\sum_{j=1}^{n} \max\left\{0, \max_{e \in \mathcal{P}_{it}}\{a_j(e)\} - \max_{e \in \mathcal{P}'_{si}}\{a_j(e)\}\right\} - \sum_{j=1}^{n} \max\left\{0, \max_{e \in \mathcal{P}_{it}}\{a_j(e)\} - \max_{e \in \mathcal{P}_{si}}\{a_j(e)\}\right\}$$

$$\leq \sum_{j=1}^{n} \max\left\{0, \max_{e \in \mathcal{P}_{si}}\{a_j(e)\} - \max_{e \in \mathcal{P}'_{si}}\{a_j(e)\}\right\}$$

By the definition of $F(\mathcal{P}, \mathcal{P}')$, the above inequality is the same as:

$$\sum_{j=1}^{n} \max\left\{0, \max_{e \in \mathcal{P}_{it}}\{a_j(e)\} - \max_{e \in \mathcal{P}'_{si}}\{a_j(e)\}\right\}$$

$$\leq \sum_{j=1}^{n} \max\left\{0, \max_{e \in \mathcal{P}_{it}}\{a_j(e)\} - \max_{e \in \mathcal{P}_{si}}\{a_j(e)\}\right\} + F(\mathcal{P}_{si}, \mathcal{P}'_{si}) \tag{24}$$

Adding inequality (24) to the assumption (23) we achieve:

$$
w_1(\mathcal{P}'_{si}) + \sum_{j=1}^{n} \max_{e \in \mathcal{P}'_{si}} \{a_j(e)\} + \sum_{j=1}^{n} \max \left\{ 0, \max_{e \in \mathcal{P}_{it}} \{a_j(e)\} - \max_{e \in \mathcal{P}'_{si}} \{a_j(e)\} \right\}
$$

$$
< w_1(\mathcal{P}_{si}) + \sum_{j=1}^{n} \max_{e \in \mathcal{P}_{si}} \{a_j(e)\} + \sum_{j=1}^{n} \max \left\{ 0, \max_{e \in \mathcal{P}_{it}} \{a_j(e)\} - \max_{e \in \mathcal{P}_{si}} \{a_j(e)\} \right\}
$$

It is easy to check that this is the same as:

$$
w_1(\mathcal{P}'_{si}) + \sum_{j=1}^{n} \max \left\{ \max_{e \in \mathcal{P}_{it}} \{a_j(e)\}, \max_{e \in \mathcal{P}'_{si}} \{a_j(e)\} \right\} < w_1(\mathcal{P}_{si}) + \sum_{i=1}^{n} \max \left\{ \max_{e \in \mathcal{P}_{it}} \{a_j(e)\}, \max_{e \in \mathcal{P}_{si}} \{a_j(e)\} \right\}
$$

which is equivalent to

$$
w_1(\mathcal{P}'_{si}) + \sum_{j=1}^{n} \max_{e \in \mathcal{P}'_{st}} \{a_j(e)\} < w_1(\mathcal{P}_{si}) + \sum_{j=1}^{n} \max_{e \in \mathcal{P}_{st}} \{a_j(e)\}
$$

By adding $w_1(\mathcal{P}_{it})$ to both sides we get the desired result $C(\mathcal{P}'_{st}) < C(\mathcal{P}_{st})$.

$\square$

Theorem 5 (Min domination) *Given a weighted directed graph $G = (V, E)$ let $w_1 : E \to \mathbb{R}$ be an additive weight function and $a_i : E \to \mathbb{R}, \quad i \in \{1, ..., n\}$ be a map from the edges of G to the reals. Then, let $w_2(\mathcal{P}) = \sum_{i=1}^{n} \min_{e \in \mathcal{P}} a_i(e)$. Let there be an objective function of the form $C(\mathcal{P}) = w_1(\mathcal{P}) - w_2(\mathcal{P})$ to be minimized. Let $\mathcal{P}_{st}$ be composed of subpaths $\mathcal{P}_{si}$ and $\mathcal{P}_{it}$ and let $\mathcal{P}'_{st}$ be composed of $\mathcal{P}'_{si}$ and $\mathcal{P}_{it}$. Let $F(\mathcal{P}, \mathcal{P}') = \sum_{j=1}^{n} \max\{0, \min_{e \in \mathcal{P}} a_j(e) - \min_{e \in \mathcal{P}'} a_j(e)\}$. Moreover let:*

$$
C(\mathcal{P}'_{si}) < C(\mathcal{P}_{si}) - F(\mathcal{P}_{si}, \mathcal{P}'_{si}) \tag{25}
$$

Then $C(\mathcal{P}'_{st}) < C(\mathcal{P}_{st})$.

In the cases where domination (25) of Theorem 5 holds we can eliminate paths $C(\mathcal{P}_{si})$ when minimizing $C(\mathcal{P})$.

Proof Clearly, since $\max_{e \in \mathcal{P}} \{a_j(e)\} = - \min_{e \in \mathcal{P}} \{-a_j(e)\}$ and therefore

$$
C(\mathcal{P}) = w_1(\mathcal{P}) - \sum_{i=1}^{n} \min_{e \in \mathcal{P}} \{-a_i(e)\} = w_1(\mathcal{P}) + \sum_{i=1}^{n} \max_{e \in \mathcal{P}} a_i(e).
$$

By Theorem 4, it follows that Theorem 5 holds.

$\square$

Theorem 4 is applied to some real life problems in Section 6.3 and 6.4. For an example of how $F(\mathcal{P}, \mathcal{P}')$ is determined see Sections 6.3 and 6.4.

5.3 Goal domination

Goal domination is a well known and commonly used way of eliminating labels (see [19]). This domination does not require monotonicity, however it can only be used for objectives $C(P)$, that are non-decreasing or non-increasing and a feasible solution to the problem must be found before goal domination can be used. The goal domination checks each new label whether it is *dominated* by a label at the destination.

To improve the goal domination one can find lower bounds (upper bounds in the case of a maximization problem) on all objectives and pairs of vertices. This can be done by preprocessing the graph data. When using preprocessing, the goal domination can be extended so that the lower

bound of the remaining path is added to the label when testing whether it is dominated by labels at the destination.

Dumitrescu and Boland [7] show that preprocessing can reduce computation time significantly on resource constrained shortest path problems. Preprocessing is useful in graphs where the edge weights seldom change. Nevertheless, the storage requirements increase as there can be a quadratic number of lower bounds. However, each lower bound will not be very space-consuming as it is a simple number. Lower bounds generated "on the fly" are often used in graphs where the vertices are placed in a coordinate system and therefore the Euclidean distance between nodes can be calculated "on the fly" [19]. Another lower bound method [19] is to make simplified versions of the graph in which a specific lower bound can be found at a given shortest path request.

6 Solving Real Life Shortest Path Problems

With the properties covered in previous section we now return to the objectives (A) through (G) listed in Section 6.1. We describe how the developed framework can be adapted to the considered objectives.

6.1 Multiplicative cost function

The multiplicative objective described in (A) is given as follows: Each edge e_{ij} has a probability $\pi_{ij} \in (0, 1]$ for being traversed successfully, and the corresponding objective is

$$C(\mathcal{P}) = \prod_{e_{ij} \in \mathcal{P}} \pi_{ij}$$

The objective is monotone since we have ([23] Example 6.3.2)

$$\max \log C(\mathcal{P}) = \max \log \prod_{e_{ij} \in \mathcal{P}} \pi_{ij} = \max \sum_{e_{ij} \in \mathcal{P}} \log \pi_{ij} \tag{26}$$

Defining $w_1 : E \to \mathbb{R}_0^+$ as $w_1(e_{ij}) = \log \pi_{ij}$ and choosing $f = e^x$, Theorem 1 gives that the objective is monotone. Therefore all subpaths are also optimal, and if the problem contains several monotone objective functions all subpaths are Pareto optimal. Notice that $w_1(e_{ij}) \geq 0$ since all $0 < \pi_{ij} \leq 1$.

6.2 Combined distance and probability function

The second objective (B) is a combination of the two criteria: distance d and probability $\pi \in (0, 1]$. The objective is written as $-d + \lambda \pi$ with $\lambda > 0$ which is to be maximized. Let $f(x) = \lambda a^x$ so that $f(\log_a(\pi(\mathcal{P}))) = \lambda \pi(\mathcal{P})$. Let $w_1(\mathcal{P}) = -d(\mathcal{P})$ and $w_2(\mathcal{P}) = \log_a(\pi(\mathcal{P}))$. As $\log_a(\pi(\mathcal{P}))$ is additive, we can use Corollary 2. Since $\log_a(\pi(\mathcal{P})) : E \to \mathbb{R}_0^-$ and f is convex, we can use Remark 2. By Remark 2 the only case where the *principle of optimality* does not work is the case where $w_2(\mathcal{P}'_{si}) > w_2(\mathcal{P}_{si})$. Therefore, we need to find a good $\mathcal{M}^-$. We choose $\mathcal{M}^- = (f(b + w_2(\mathcal{P}_{si})) - f(b + w_2(\mathcal{P}'_{si})))/(w_2(\mathcal{P}_{si}) - w_2(\mathcal{P}'_{si}))$ where $b \leq w_2(\mathcal{P}_{it})$ for all simple paths $\mathcal{P}_{it}$ from i to t on G as suggested in Remark 2 Equation (18). Inserting this value into the definition of dominance for the case $w_2(\mathcal{P}'_{si}) > w_2(\mathcal{P}_{si})$ in Corollary 2 we get:

$$w_1(\mathcal{P}'_{si}) > w_1(\mathcal{P}_{si}) + f(b + w_2(\mathcal{P}_{si})) - f(b + w_2(\mathcal{P}'_{si})) \tag{27}$$

A lower bound L for $\pi(\mathcal{P}_{it})$ where $\mathcal{P}_{it}$ is an arbitrary simple path from i to t on G results in $\log_a(L)$ being a lower bound for $\log_a(\pi(\mathcal{P}_{it}))$. Therefore we can substitute b with $\log_a(L)$. Moreover, we can use that $f(x) = \lambda a^x$, $w_1(\mathcal{P}) = -d(\mathcal{P})$ and $w_2(\mathcal{P}) = \log_a(\pi(\mathcal{P}))$ to rewrite (27) as:

$$w_1(\mathcal{P}'_{si}) > w_1(\mathcal{P}_{si}) + \lambda a^{\log_a(L) + \log_a(\pi(\mathcal{P}_{si}))} - \lambda a^{\log_a(L) + \log_a(\pi(\mathcal{P}'_{si}))}$$

this inequality is the same as

$$w_1(\mathcal{P}'_{si}) > w_1(\mathcal{P}_{si}) + \lambda L\pi(\mathcal{P}_{si}) - \lambda L\pi(\mathcal{P}'_{si}),$$

and hence

$$-d(\mathcal{P}'_{si}) > -d(\mathcal{P}_{si}) + \lambda L\pi(\mathcal{P}_{si}) - \lambda L\pi(\mathcal{P}'_{si})). \tag{28}$$

This means that the dominance of Corollary 2 in this case is: If (29) or (30) is satisfied then then $C(\mathcal{P}'_{st}) > C(\mathcal{P}_{st})$.

$$-d(\mathcal{P}'_{si}) + \pi(\mathcal{P}'_{si}) > -d(\mathcal{P}_{si}) + \pi(\mathcal{P}_{si}) \quad \text{and} \quad \pi(\mathcal{P}'_{si}) \le \pi(\mathcal{P}_{si}) \tag{29}$$

$$-d(\mathcal{P}'_{si}) > -d(\mathcal{P}_{si}) + \lambda L\pi(\mathcal{P}_{si}) - \lambda L\pi(\mathcal{P}'_{si})) \quad \text{and} \quad \pi(\mathcal{P}'_{si}) > \pi(\mathcal{P}_{si}) \tag{30}$$

Carraway et al. [4] propose the following algorithm for implementing the dominance test. Let P be a path from a vertex s to a vertex i then d is the distance of the path P and π is the probability of the path P. Let the path P' be a path from s to i different from P and let $\hat{d}$ and $\hat{\pi}$ be the distance and probability on that path.

Algorithm by Carraway et al.:

Step 0. Designate (d, π) and $(\hat{d}, \hat{\pi})$ such that $d \le \hat{d}$.

Step 1. If $d = \hat{d}$ and $\pi \le \hat{\pi}$ then delete (d, π) and stop; else if $d = \hat{d}$ and $\pi > \hat{\pi}$ then delete $(\hat{d}, \hat{\pi})$ and stop.

Step 2. If $\pi = \hat{\pi}$, delete $(\hat{d}, \hat{\pi})$ and stop.

Step 3. If $\hat{d} - d \le \lambda b_j(\hat{\pi} - \pi)$ where b_j is the minimum of the probability on the remaining path. Delete (d, π) and stop.

Step 4. If $d - \hat{d} \le \lambda m_j(\pi - \hat{\pi})$ where m_j is the maximum of the probability on the remaining path. Delete $(\hat{d}, \hat{\pi})$ and stop.

Step 5. Retain both returns and stop.

In [4] Step 2 was written as follows: *If $\pi \le \hat{\pi}$, delete $(\hat{d}, \hat{\pi})$ and stop.* As this does not hold, it is presumed to be a typing error and we have inserted $=$ instead of $\le$. Note that by using a non-strict inequality, Carraway et al. only get the *minimal complete* set of solutions. Clearly, by making the inequality non-strict, our dominance method, (29) and (30), will find exactly the *minimal complete* set of Pareto optimal solutions.

We now show that Theorem 2 allows for the elimination of all of the paths eliminated by the algorithm of Carraway et al.

It can easily be shown that all paths deleted by step 1 and 2 are eliminated by Corollary 2, inequalities (29) and (30).

In order to prove the same for step 3 we have to show that when $d \le \hat{d}$ and $\hat{d} - d \le \lambda b_j(\hat{\pi} - \pi)$, where b_j is the minimum of the probabilities, on the remaining path $(\pi(P_{it}))$, then one of the domination statements (29) and (30) holds. Let (d, π) be the pair $(d(P_{si}), \pi(P_{si}))$ and $(\hat{d}, \hat{\pi})$ be the pair $(d(P'_{si}), \pi(P'_{si}))$. Since $b_j > 0$ we can choose the lower bound $L = b_j > 0$. When $\hat{\pi} < \pi$ no path is deleted in step 3. Now insert the values in the inequality of step 3:

$$d(P'_{si}) - d(P_{si}) \le \lambda L(\pi(P'_{si}) - \pi(P_{si})) \quad \Rightarrow \quad -d(P'_{si}) + d(P_{si}) \ge \lambda L(-\pi(P'_{si}) + \pi(P_{si})) \tag{31}$$

$$\Rightarrow \quad -d(P'_{si}) \ge -d(P_{si}) + \lambda L(\pi(P_{si}) - \pi(P'_{si})) \tag{32}$$

which by (30) means that Corollary 2 eliminates (d, π) when $\hat{\pi} > \pi$. In the case where $\hat{\pi} = \pi$ it is easy to show that Corollary 2 eliminates (d, π).

To show that a path deleted in step 4 is also deleted by (29) and (30), set $(d, \pi) = (d(\mathcal{P}'_{si}), \pi(\mathcal{P}'_{si}))$ and $(\hat{d}, \hat{\pi}) = (d(\mathcal{P}_{si}), \pi(\mathcal{P}_{si}))$ and note that in the case where $\pi > \hat{\pi}$, the inequality (30) is trivially true. If $\pi \le \hat{\pi}$, then we must show that the inequality of step 4 implies (29). But this must be true as (29) amounts to the principle of optimality (regular domination).

6.3 Maximum of commissions

The objective described in (C) is taken from [2]. The cost of a path is calculated as the price of the edges (transports) plus some commission to agents.

Let the set of agents be $1, \ldots, A$ and let S_a be the edges covered by agent $a \in \{1, \ldots, A\}$. Each edge is covered by at most one agent. Each edge e_{ij} has a corresponding commission c_{ij}^a. An agent is paid the largest commission he is entitled to on the path $\mathcal{P}$. The objective is then

$$C(\mathcal{P}) = \sum_{e_{ij} \in \mathcal{P}} c_{ij} + \sum_{a=1}^{A} \max_{e_{ij} \in S_a \cap \mathcal{P}} \{0, c_{ij}^a\}$$

In this case we can use Theorem 4 to obtain that

$$\sum_{e_{ij} \in \mathcal{P}_{sh}'} c_{ij} + \sum_{a=1}^{A} \max_{e_{ij} \in S_a \cap \mathcal{P}_{sh}'} \{0, c_{ij}^a\} < \sum_{e_{ij} \in \mathcal{P}_{sh}} c_{ij} + \sum_{a=1}^{A} \max_{e_{ij} \in S_a \cap \mathcal{P}_{sh}} \{0, c_{ij}^a\} - F(\mathcal{P}_{sh}, \mathcal{P}_{sh}')$$

where

$$F(\mathcal{P}_{sh}, \mathcal{P}_{sh}') = \sum_{a=1}^{A} \max\{0, \max_{e_{ij} \in S_a \cap \mathcal{P}_{sh}} \{0, c_{ij}^a\} - \max_{e_{ij} \in S_a \cap \mathcal{P}_{sh}'} \{0, c_{ij}^a\}\}$$

implies $C(\mathcal{P}') < C(\mathcal{P})$.

6.4 Number of zones visited

In the *zone system* objective described in (D) the price is calculated depending entirely on the zones passed on the journey. More formally let the set of edges be divided into zones $Z_1, \ldots, Z_n$. Without loss of generality we may assume that each edge only corresponds to one zone, as we otherwise may split the edge. Let w_e have the value 1 for all edges e.

The objective then is to minimize the number of different zones on a path $\mathcal{P}$. The objective can be written as follows:

$$\text{min} : C(\mathcal{P}) = \sum_{h=1}^{n} \max_{e \in Z_h \cap \mathcal{P}} \{0, w_e\} \tag{33}$$

To use Theorem 4 on the zone system each edge must be assigned the zones it travels through. Let

$$F(\mathcal{P}, \mathcal{P}') = \sum_{h=1}^{n} \max\{0, \max_{e \in Z_h \cap \mathcal{P}} \{0, w_e\} - \max_{e \in Z_h \cap \mathcal{P}'} \{0, w_e\}\}$$

By Theorem 4 we get that

$$\sum_{h=1}^{n} \max_{e \in Z_h \cap \mathcal{P}_{si}'} \{0, w_e\} < \sum_{h=1}^{n} \max_{e \in Z_h \cap \mathcal{P}_{si}} \{0, w_e\} - F(\mathcal{P}_{si}, \mathcal{P}_{si}') \tag{34}$$

implies that the $C(\mathcal{P}_{st}') < C(\mathcal{P}_{st})$.

However for objectives of the form $\sum_{h=1}^{n} \max_{e \in Z_h \cap \mathcal{P}} \{0, w_e\}$ where $w_1(\mathcal{P})$ is 0 for all paths, Theorem 4 will not be able to eliminate any simple subpaths from the investigation. This means that in this case no subpath will be eliminated by the dominance of Theorem 4. In this case *goal domination* must be used to eliminate paths that are not Pareto optimal compared to a solution that is already found. This, however, requires that a solution is found. Another way to circumvent the problem is to combine an additive criterion with the price in the objective so that $w_1(\mathcal{P})$ will have a value greater than zero. The other criterion could for example be time or distance.

6.5 Maximum zone distance from origin

The objective (E) of finding the maximum zone distance from a origin can be handled by creating new zones around the source s. Let Z_1' be the zone in which s is located. Let Z_2' be the set of zones that are adjacent to Z_1'. In general Z_h' is the set of zones adjacent to Z_{h-1}'. Note that to get to the distance of three zones of the origin one must have passed zones distance two and one from origin. Let the cost function $c_{ij} = c_h$ where $e_{ij} \in Z_h'$ so that h is the number of zones visited. In this case the objective function is:

$$C(\mathcal{P}) = \max_{e_{ij} \in Z' \cap \mathcal{P}} \{c_{ij}\} \tag{35}$$

Clearly in this case if

$$\max_{e_{ij} \in Z' \cap \mathcal{P}_{si}'} \{c_{ij}\} \leq \max_{e_{ij} \in Z' \cap \mathcal{P}_{si}} \{c_{ij}\} \tag{36}$$

then $C(\mathcal{P}_{st}') \leq C(\mathcal{P}_{st})$. Notice that it is possible for the domination to eliminate paths if only the *minimal complete* set is desired. If all paths of minimum value are desired then *goal domination* must be used.

6.6 Zone distance and time

The objective (F) is another version of objective (E). However this objective takes a time and a number of zones and returns a cost. With reference to the Copenhagen ticket system, we assume that the cost function is the number of time intervals times a factor σ. Let us suppose that the time intervals all are of some size k. The time is clearly additive and, as in the model of (E), the zones are not. It is a general assumption that the cost function is monotonously increasing. Moreover the time and zone criteria are not independent in the objective function as the cost is the maximum of the cost of the ticket for the time used and the cost of the ticket for the traveled number of zones. The objective now becomes:

$$C(\mathcal{P}) = \max \left\{ c \left(\sum_{e_{ij} \in \mathcal{P}} t_{ij} \right), \max_{e_{ij} \in Z' \cap \mathcal{P}} \{c_{ij}\} \right\} \tag{37}$$

From the dominance of objective (E) we have that:

$$\max_{e_{ij} \in Z' \cap \mathcal{P}_{si}} \{c_{ij}\} \leq \max_{e_{ij} \in Z' \cap \mathcal{P}_{si}'} \{c_{ij}\} \Rightarrow \max_{e_{ij} \in Z' \cap \mathcal{P}_{st}} \{c_{ij}\} \leq \max_{e_{ij} \in Z' \cap \mathcal{P}_{st}'} \{c_{ij}\} \tag{38}$$

Since the cost function is the product of a factor σ and the number of time intervals, and all time intervals are of same size k, we get

$$c \left(\sum_{e_{ij} \in \mathcal{P}} t_{ij} \right) = \sigma \left\lceil \frac{1}{k} \sum_{e_{ij} \in \mathcal{P}} t_{ij} \right\rceil \tag{39}$$

It is easy to see that:

$$\sigma \left\lceil \frac{a+c}{k} \right\rceil - \sigma \left\lceil \frac{a}{k} \right\rceil + \sigma \geq \sigma \left\lceil \frac{b+c}{k} \right\rceil - \sigma \left\lceil \frac{b}{k} \right\rceil \tag{40}$$

When $a \bmod k \geq b \bmod k$ and for all other cases regular domination holds.

For two paths $\mathcal{P}$ and $\mathcal{P}'$, we now by Theorem 3 have that:

$$\sigma \left\lceil \frac{1}{k} \sum_{e_{ij} \in \mathcal{P}_{si}'} t_{ij} \right\rceil - \sigma \geq \sigma \left\lceil \frac{1}{k} \sum_{e_{ij} \in \mathcal{P}_{si}} t_{ij} \right\rceil \Rightarrow \sigma \left\lceil \frac{1}{k} \sum_{e_{ij} \in \mathcal{P}_{st}'} t_{ij} \right\rceil \geq \sigma \left\lceil \frac{1}{k} \sum_{e_{ij} \in \mathcal{P}_{st}} t_{ij} \right\rceil \tag{41}$$

holds when $\sum_{e_{ij} \in \mathcal{P}'_{si}} t_{ij} \bmod k \geq \sum_{e_{ij} \in \mathcal{P}_{si}} t_{ij} \bmod k$. Let $\alpha = \sigma$ if $\sum_{e_{ij} \in \mathcal{P}'_{si}} t_{ij} \bmod k \geq \sum_{e_{ij} \in \mathcal{P}_{si}} t_{ij} \bmod k$, and $\alpha = 0$ otherwise. Clearly, if:

$$\sigma \left\lceil \frac{1}{k} \sum_{e_{ij} \in \mathcal{P}_{si}} t_{ij} \right\rceil \leq \sigma \left\lceil \frac{1}{k} \sum_{e_{ij} \in \mathcal{P}'_{si}} t_{ij} \right\rceil - \alpha \tag{42}$$

and

$$\max_{e_{ij} \in Z' \cap \mathcal{P}_{si}} \{c_{ij}\} \leq \max_{e_{ij} \in Z' \cap \mathcal{P}'_{si}} \{c_{ij}\} \tag{43}$$

then $C(\mathcal{P}_{st}) \leq C(\mathcal{P}'_{st})$.

6.7 Modulo k penalties

Let S be a given set of edges. We consider a shortest path problem where we pay an additional penalty $\sigma \geq 0$ each time the set S has been visited k times. Our objective is hence

$$C(\mathcal{P}) = \sum_{e_{ij} \in \mathcal{P}} w_{ij} + \sigma \left\lfloor \frac{1}{k} \sum_{e_{ij} \in S \cap \mathcal{P}} c_{ij} \right\rfloor = W(\mathcal{P}) + \sigma \left\lfloor \frac{1}{k} C_S(\mathcal{P}) \right\rfloor \tag{44}$$

Jepsen et al. [16] handle this problem by using a dominance criterion that takes into account when the cost $\sigma \lfloor c_{ij}/k \rfloor$ is to be paid. In other words, if two subpaths $\mathcal{P}_{si}$ and $\mathcal{P}'_{si}$ end at the same node i and either (45) or (46) is satisfied then $\mathcal{P}'_{si}$ dominates $\mathcal{P}_{si}$.

$$C(\mathcal{P}'_{si}) + \sigma \leq C(\mathcal{P}_{si}) \quad \text{and} \quad C_S(\mathcal{P}'_{si}) \bmod k \geq C_S(\mathcal{P}_{si}) \bmod k \tag{45}$$
$$C(\mathcal{P}'_{si}) \leq C(\mathcal{P}_{si}) \quad \text{and} \quad C_S(\mathcal{P}'_{si}) \bmod k \leq C_S(\mathcal{P}_{si}) \bmod k \tag{46}$$

We have previously seen that:

$$\sigma \left\lfloor \frac{a+c}{k} \right\rfloor - \sigma \left\lfloor \frac{a}{k} \right\rfloor - \sigma \leq \sigma \left\lfloor \frac{b+c}{k} \right\rfloor - \sigma \left\lfloor \frac{b}{k} \right\rfloor \tag{47}$$

holds when $a \bmod k \geq b \bmod k$ and for all other cases regular domination holds.

By Theorem 3 we now have that:

$$W(\mathcal{P}_{si}) + \sigma \left\lfloor \frac{1}{k} C_S(\mathcal{P}_{si}) \right\rfloor + \sigma \leq \sigma \left\lfloor \frac{1}{k} W(\mathcal{P}'_{si}) + C_S(\mathcal{P}'_{si}) \right\rfloor$$
$$\Rightarrow W(\mathcal{P}_{st}) + \sigma \left\lfloor \frac{1}{k} C_S(\mathcal{P}_{st}) \right\rfloor \leq W(\mathcal{P}'_{st}) + \sigma \left\lfloor \frac{1}{k} C_S(\mathcal{P}'_{st}) \right\rfloor \tag{48}$$

holds when $C_S(\mathcal{P}_{si}) \bmod k \geq C_S(\mathcal{P}'_{si}) \bmod k$ and otherwise regular domination holds. Since Theorem 3 was inspired by the domination found in Jepsen et al. [16] it is not a surprise that it holds for their case.

7 Computational Results

In this section we demonstrate that even though non-additive criteria functions are hard to solve in theory, several real-life problems are tractable in practice. The label-correcting algorithm was implemented in C++ and all tests were carried out on a 2.1 GHz Pentium processor.

The considered instances are based on real-life data from a shipping company which wants to find the Pareto optimal paths for transporting a container from s to t when considering various

objectives. The given network data contained 15 vertices and 125 edges. In order to construct larger instances, the network has been upscaled.

In all tests we have restricted the number of transfers to at most 10. This number is quite large and in reality could be set lower. For comparison, we have run tests where the Pareto optimal solutions are found without using the nonadditive domination. In some of the graph instances the tests not using nonadditive domination while using goal dominance only were omitted from the test set as they were too time consuming.

To improve the performance of the algorithm we have applied some extra methods for eliminating undesirable paths such as *goal dominance* described in Section 5.3. In the case of *goal dominance* the standard elimination is used for all objectives. Another way to improve elimination is to preprocess the data by creating lower bounds between all pairs of vertices for all objectives. This method was described in Section 5.3. The lower bounds can be used to see if a subpath will be sure to yield a *dominated* path. This process uses the subpath values combined with the lower bounds of all objectives from the current vertex to the destination to check for dominance against a path already found.

Transfer, time and price with maximum of commissions objectives

Instance	Number of Pareto optimal solutions found on 100 random requests	Time for preprocessing (s)	Number of hubs	Number of departures
R1	571	3.96	91	1176
R2	491	137.42	221	4176
R3	583	2934.72	321	10176

	With weighted domination									Without weighted domination					
	No optimization			Goal dominance			Lower bounds			Goal dominance			Lower bounds		
	R1	R2	R3	R1	R2	R3	R1	R2	R3	R1	R2	R3	R1	R2	R3
Avg. time (s)	0.24	0.94	4.30	0.12	0.40	1.98	0.08	0.23	1.05	15.74	-	-	3.71	3.87	7.05
Max. time (s)	0.54	1.72	9.23	0.39	1.30	7.11	0.25	0.91	4.84	543.04	-	-	243.25	141.82	147.18
Avg. # labels	1.6e5	6.9e5	2.4e6	8.0e4	3.0e5	1.2e6	5.1e4	1.6e5	5.3e5	4.2e5	-	-	1.7e5	4.6e5	1.1e6

Table 1: Running times and number of labels generated when finding the Pareto optimal paths in a graph. The tests are done with 100 random requests where one of the objectives contains a max function.

Transfer, time and price with maximum of commissions and probability objectives

Instance	Number of Pareto optimal solutions found on 100 random requests	Time for preprocessing (s)	Number of hubs	Number of departures
R1	652	0.24	21	176
R2	3665	9.90	91	1176
R3	3242	55.15	121	2176

	With weighted domination									Without weighted domination					
	No optimization			Goal dominance			Lower bounds			Goal dominance			Lower bounds		
	R1	R2	R3	R1	R2	R3	R1	R2	R3	R1	R2	R3	R1	R2	R3
Avg. time (s)	0.06	7.18	15.60	0.04	2.19	6.66	0.02	0.46	1.01	3.21	-	-	0.07	34.41	55.02
Max. time (s)	0.18	32.83	70.04	0.18	19.17	61.33	0.08	4.77	14.26	130.21	-	-	1.08	2244.09	2174.12
Avg. # labels	3.7e4	9.9e5	2.4e6	2.3e4	4.4e5	1.1e6	1.0e4	1.2e5	2.6e5	1.4e5	-	-	2.3e4	4.5e5	9.0e6

Table 2: Running times and number of labels generated when finding the Pareto optimal paths in a graph. The tests are done with 100 random requests where one of the objectives is the probability function.

The tests reported in Table 1, considers a real-life shipping problem similar to case C described in Section 6.1. The shipping company wants to find the Pareto optimal paths for transporting a container from s to t when considering the time, the number of transfers, and the cost. The cost is non-additive as it includes payments to agents along the path. Even though the same agent may

Time and price with maximum of commissions and transfer-probability objectives

Instance	Number of Pareto optimal solutions found on 100 random requests	Time for preprocessing (s)	Number of hubs	Number of departures
R1	408	0.23	21	176
R2	700	9.90	91	1176
R3	799	54.12	121	2176

	With weighted domination									Without weighted domination					
	No optimization			Goal dominance			Lower bounds			Goal dominance			Lower bounds		
	R1	R2	R3	R1	R2	R3	R1	R2	R3	R1	R2	R3	R1	R2	R3
Avg. time (s)	0.04	0.37	1.40	0.02	0.16	0.55	0.01	0.10	0.34	0.81	-	-	0.08	1.76	6.19
Max. time (s)	0.11	0.84	9.44	0.08	0.69	4.06	0.08	0.38	2.33	28.50	-	-	1.82	76.89	146.86
Avg. # labels	2.0e4	2.1e5	4.7e5	1.2e4	1.0e5	2.1e5	7.9e3	5.9e4	1.3e5	5.5e4	-	-	2.0e4	1.8e5	3.9e5

Table 3: Running times and number of labels generated when finding the Pareto optimal paths in a graph. The tests are done with 100 random requests where one of the objectives is the probability combined with transfer function.

be responsible for several vertices on a path, the agent only gets paid once, corresponding to the largest commission the agent is entitled to along the path taken. To solve this problem we have used *max-dominance* (Theorem 4) for the cost objective and regular *dominance* (Theorem 1) for the time and transfer objectives.

To illustrate the strength of the domination presented in this paper we have compared the results with results where we only eliminate a path at an intermediate point if all the commissions to agents on the eliminated path are smaller than or equal to the commissions paid on the dominating path. Note that the random requests are different for the different test instances. The fact that the requests are different for the instances may explain the surprising result that the longest time it takes to solve a problem without using *max-dominance* takes longer for the smaller $R1$ instance than for the other two larger instances.

In the second test, reported in Table 2, the four objectives of time, transfers, cost and the probability of reaching the destination are considered. The probability objective is described in case A of Section 6.1. As we did not have access to real-life probability data, the probability of each edge was uniformly randomly distributed. It is clear that with an additional objective the complexity of the problem will increase. This can be seen from the fact that the average number of Pareto optimal paths in Table 2 for instance $R2$ and $R3$ is more than 32 per request, where real-life instances typically have fewer. Note that the probability objective uses regular domination as it has the monotonicity property. However the price objective is the nonadditive objective from case C described in Section 6.1. To solve this problem we have used *max-dominance* (Theorem 4) for the cost objective and regular *dominance* (Theorem 1) for the time, transfers and probability objectives. In the comparison without using nonadditive domination we have changed the nonadditive domination on the price objective as in Table 1. It is clear that the *max-dominance* has a significant impact on the running time as does the lowerbound method even though the the time used on preprocessing increases significantly with the instance size.

In the third test, reported in Table 3, the three objectives time, price and the weighted objective of transfers and probability was considered. The weighted function of an additive function and the probability function is described for the distance and probability in case B of Section 6.1. The price objective is the nonadditive objective from case C described in Section 6.1. To solve this problem we have used Corollary 2 for the weighted transfer and probability objective and regular *dominance* (Theorem 1) for the time objective and the *max-dominance* (Theorem 4) for the price objective.

For comparison tests reported in Table 1, 2 and 3 are also run without using the domination of Corollary 2. In that case the weighted objective of transfer and probability is dominated when the transfers are greater and the probability is smaller than those on the dominating path. For the time objective we use ordinary domination, while the price objective makes use of the same

domination as in the previous tests.

The results clearly indicate that the presented tightened dominance methods for non-additive objectives significantly decrease the number of labels generated. In fact for the large graphs, less than a third of the labels are generated when using the non-additive domination methods for objectives. The data used for the lower bound method generated during preprocessing can be used for all requests as long as the graph and objectives are the same. The time used on preprocessing clearly depends on the graph size. If the graph seldom is changed, it is computationally cheap to find the lower bounds. In the case of Table 2 instance $R3$ it is evident that preprocessing reduces the running time and number of generated labels significantly.

Not surprisingly, comparing instances $R2$ and $R3$ in Tables 2 and 3 it is clear that adding objectives has an impact on the complexity of the problem with an significant increase in the number of Pareto optimal solutions. For all the tests, the tightened domination reduces the average running time by at least a factor three and the longest running time by at least a factor two. One can also see that the improvement increases as the graph gets larger.

8   Conclusion

This paper has presented some general techniques to restrict the subpaths that need to be investigated in a dynamic programming algorithm, when solving shortest path problems with several non-additive functions. The dynamic programming method for the shortest path problem is simple and easy to adjust when the complicated cost functions that arise in real-life applications are encountered. The domination criteria presented in the theorems can, as shown in Section 6, be used on different real-life applications. However it should be noted that there are still many non-additive criteria and objectives not covered by the theorems presented in this article.

Based on the experimental results we may conclude that the tightened domination method significantly lowers the number of labels generated for each problem. Using better data structures which make it possible to quickly test for domination may further improve the running times.

Being able to handle non-additive objectives, the shortest path algorithms can more widely be applied to real-life problems. We believe that the tightened domination criteria given in this paper can be a tool box for others when defining such criteria for other real-life problems. Our method is a contribution in the process of optimizing more and more real-life problems and thereby using our resources optimally and minimizing waste both in the literal and metaphorical sense.

References

[1] R. Ahuja, T. Magnanti, and J. Orlin. *Network Flows : Theory, Algorithms, and Applications.* Prentice Hall, 1993.

[2] L. Blander-Reinhardt. Multi-objective shortest path for cargo transportation. Master's thesis, University of Copenhagen, DIKU, Denmark, April 2005.

[3] J. Brumbaugh-Smith and D. Shier. An empirical investigation of some bicriterion shortest path algorithms. *European Journal of Operational Research*, 43:216–224, 1989.

[4] R. Carraway, T. L. Morin, and H. Moskowitz. Generalized dynamic programming for multi-criteria optimization. *European Journal of Operational Research*, 44:95–104, 1990.

[5] T. Cormen, C. Leiserson, and R. Rivest. *Introduction to Algorithms*. The MIT Press/McGraw-Hill Book Company, 1997.

[6] G. Desaulniers, J. Desrosiers, I. Ioachim, M. Solomon, F. Soumis, and D. Villeneuve. *Fleet Mangaement and Logistics*. Kulwer Academic Publisher, 1998. chapter: Unified Framework for Deterministic Time Constrained Vehice Routing and Crew Scheduling Problems.

[7] I. Dumitrescu and N. Boland. Improved preprocessing, labeling and scaling algorithms for the weight-constrained shortest path problem. *Networks*, 42:135–153, 2003.

[8] M. Ehrgott and X. Gandibleux. An annotated bibliography of multiobjective combinatorial optimization. *OR Spectrum*, 22:425–460, 2000.

[9] C. Gueguen, P. Dejax, M. Dror, and M. Gendreau. An exact algorithm for the elementary shortest path problem with resource constraints. Technical report, Laboratoire Productique Logistique, Ecole Centrale Paris, 1998. revised in July 1999, also with D. Feillet as co-author.

[10] P. Hansen. Bicriteria path problems. *G. Fandel and T. Gal: Multi Criteria Decision Making Theory and Applications, Lecture Notes in Economics and Mathematical Systems*, 177:109–127, 1979.

[11] HUR. http://trafikinfo.hur.dk/priserogbilletter.

[12] K. Hutson and D. Shier. Extended dominance and a stochastic shortest path problem. *Computers & Operations Research*, 36:584–596, 2009.

[13] I. Ioachim, S. Gelinas, J. Desrosiers, and F. Soumis. A dynamic programming algorithm for the shortest path problem with time windows and linear node costs. *Networks*, 31:193–204, 1998.

[14] S. Irnich and G. Desaulniers. Shortest path problems with resource constraints. In M. M. S. G. Desaulniers, J. Desrosiers, editor, *Column Generation*, pages 33–65. Springer, USA, 2005.

[15] S. Irnich and D. Villeneuve. The shortest path problem with resource constraints and k-cycle elimination for $k > 3$. *Informs Journal on Computing*, 18:391–406, 2006.

[16] M. Jepsen, B. Petersen, S. Spoorendonk, and D. Pisinger. Subset-row inequalities applied to the vehicle-routing problem with time windows. *Operations Research*, 56:497–511, 2008.

[17] T. Lengauer and D. Theune. Efficient algorithms for path problems with gernal cost citeria. In *International Colloquium on Automata, Languages and Programming (ICALP)*, pages 314 – 326, 1991.

[18] E. Martins. On a multicriteria shortest path problem. *European Journal of Operation Research*, 16:236–237, 1984.

[19] M. Müller-Hannemann and M. Schnee. Finding all attractive train connections by multi-criteria pareto search. In *Proceedings of the 4th Workshop on Algorithmic Methods and Models for Optimization of Railways (ATMOS 2004) Bergen, Norway, Lecture Notes in Computer Science*, 2004.

[20] M. Müller-Hannemann and K. Weihe. On the cardinality of the pareto set in bicriteria shortest path problems. *Annals of Operation Research*, 147:185–197, 2004.

[21] W. Powell and Z. Chen. A generalized threshold algorithm for the shortest path problem with time windows. In P. Pardalos and D. Du, editors, *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pages 303–318. American Mathematical Society, 1998.

[22] S. Skriver and K. Andersen. A label correcting approach for solving bicriterion shortest-path problems. *Computers & Operations Research*, 27:507–524, 2000.

[23] H. Taha. *Operations Research - An introduction.* Pearson Prentice Hall, eight edition edition, 2007.

[24] G. Tsaggouris and C. Zaroliagis. Non-additive shortest paths. In *Algorithms - ESA 2004. 12th Annual European Symposium*, volume 3221 of *Lecture Notes in Computer Science*, pages 822–834, 2004.

[25] G. Tsaggouris and C. Zaroliagis. Multiobjective optimization: improved fptas for shortest paths and non-linear objectives with applications. In *Algorithms and Computation. 17th International Symposium, ISAAC 2006*, volume 4288 of *Lecture Notes in Computer Science*, pages 389–398, 2006.

[26] D. Villeneuve and G. Desaulniers. The shortest path problem with forbidden paths. *European Journal of Operational Research*, 165:97–107, 2005.

Shortest path problems appear as subproblems in numerous optimization problems. In most papers concerning multiple objective shortest path problems, additivity of the objective is a de-facto assumption, but in many real-life situations objectives and criteria, can be non-additive. The purpose of this paper is to give a general framework for dominance tests for problems involving a number of non-additive criteria. These dominance tests can help eliminate paths in a dynamic programming framework when using multiple objectives. Results on real-life multi-objective problems containing non-additive criteria are reported. We show that in many cases the framework can be used to efficiently reduce the number of generated paths.