Soft Computing based Optimisation and Decision Models for Adaptive Maps

Marina Torres Anaya

Advisor: David Alejandro Pelta

UNIVERSIDAD DE GRANADA

Departamento de Ciencias de la Computación e Inteligencia Artificial

Phd program: Information and Communication Technologies

October 2019

Editor: Universidad de Granada. Tesis Doctorales
Autor: Marina Torres Anaya
ISBN: 978-84-1306-415-4
URI: http://hdl.handle.net/10481/58823

Agradecimientos

Este trabajo ha sido posible gracias a la Universidad de Granada y gracias al Ministerio de Economía y Competitividad por la concesión de una ayuda para contratos predoctorales para la formación de doctores cofinanciada por el Fondo Social Europeo, con referencia BES-2015-073429, asociada al proyecto TIN2014-55024-P.

Estos últimos años han sido intensos. Intensos en los momentos de éxito pero también algo intensos en los momentos más complicados. Sin embargo, no he estado sola en este viaje. He podido celebrar y disfrutar todos los logros alcanzados en buena compañía y he podido contar con el apoyo de mucha gente cuando me he visto más perdida. A todos los que me habéis acompañado, os lo agradezco enormemente.

Tengo claro que estos años no habrían dado su fruto sin el apoyo de mi familia. Muchas gracias por haber celebrado conmigo los triunfos y por ayudarme cuando más lo he necesitado.

Marina y Migue, habéis sido dos pilares fundamentales en esta etapa de mi vida. Gracias por hacerme el camino más llevadero con vuestra compañía. Me siento muy afortunada por contar con vuestra amistad.

Muchas gracias a David por haber sido mi director y tutor, haberme enseñado tanto y haberme ayudado a crecer no solo en el ámbito académico, si no también en el personal.

Tengo mucho que agradecer al grupo MODO, especialmente a Curro, por las incontables experiencias positivas que me han aportado y por haberme hecho sentir una más en la familia. Por todo el apoyo académico y personal que me habéis dado y por brindarme la posibilidad de participar en el mundo de la investigación yendo de la mano de grandes investigadores y divulgadores como vosotros, gracias.

I would like to thank ICOS research group for giving me the opportunity to work with them. Natalio, thank you very much for making this possible and making me feel part of the group from my first day. Shouyong, thanks for your time and your teachings. I have very good memories of working with you. Xue, Ben, Nishant, Chris, Nunzia and Frances, I will always remember with joy the time I spent with all of you. Thank you very much.

A mis abuelos . . .

Table of contents

Abstract

This thesis has been developed within a research training staff grant associated with the project TIN2014-55024-P from the Spanish Ministry of Economy and Competitiveness and co-funded by the European Social Fund. This project raises as initial hypothesis the fact that decision and optimisation models can be more efficiently addressed by using Soft Computing techniques and, consequently, better solutions can be found for the corresponding social challenges. Within the three scenarios proposed in the project, this thesis is oriented to the study of "adaptive maps".

Maps and maps based applications are useful and generally known. However, the uncertainty and imprecision of the real world characteristics is usually not considered. Moreover, route planning applications usually provide solutions with independence of the user's preferences, despite each person has a personal view of the world having different priorities and needs. For example, a person with some mobility reduction could consider

that a really short route full of stairs is non-viable. That same situation is completely different for a person that can easily use the stairs, who maybe prefers that route instead of a longer one.

As everyone sees the world in different way, there is no unique solution that can fit everyone's needs. That implies that when finding proper routes on a map, like pedestrian ones, it is crucial to understand who the final user is. The preferences (their modelling and proper management) are the key to define what is important for each individual.

The objective for this work is the use of Soft Computing for the improvement and resolution of decision and optimisation models on the adaptive maps scenario. The objective is divided into three sub-objectives: define a model to solve the personalised route problem and provide a solution approach, define a model for adaptive maps and provide a solution approach and define a posteriori method to select routes by incorporating preferences after the optimisation process.

For that purpose, the personalised route problem is first presented in detail, a model is proposed and a solution approach is provided. Secondly, the concept of adaptive maps is presented, discussed and developed. In short, an adaptive map is a map able to show how an user sees the world not only in terms of distances but also considering other related characteristics whose importance is determined by that user.

The attainment of best solutions for the personalised route problem (in terms of the user's preferences) can also be explored

using a different and more general approach: the selection of solutions of interest. This a posteriori method consists in selecting personalised routes starting from a wide set of routes, obtained after solving a multiobjective problem, which will be later evaluated taking into account the user's preferences, thus identifying the solutions that are most interesting for current user. Such approach is considered and explored in this work by defining the solution of interest identification problem and providing a solution approach.

Resumen

Esta tesis se ha desarrollado con una ayuda para contratos predoctorales para la formación de doctores, concedida por el Ministerio de Economía y Competitividad y cofinanciada por el Fondo Social Europeo, asociada al proyecto TIN2014-55024-P. Este proyecto plantea como hipótesis inicial el hecho de que los modelos de decisión y optimización pueden abordarse de manera más eficiente mediante el uso de técnicas de Soft Computing y, en consecuencia, se pueden encontrar mejores soluciones para los desafíos sociales correspondientes. Dentro de los tres escenarios propuestos en el proyecto, esta tesis está orientada al estudio de los "mapas adaptativos".

Los mapas y las aplicaciones basadas en mapas son útiles y muy conocidos. Sin embargo, no se suele considerar la incertidumbre e imprecisión típicas de las características del mundo real. Además, las aplicaciones de rutas suelen proporcionar soluciones independientemente de las preferencias del usuario, a pesar de que cada persona tiene una visión personal del mundo

con prioridades y necesidades diferentes a las de los demás. Por ejemplo, una persona con movilidad reducida podría considerar no es viable una ruta realmente corta llena de escaleras. Esa misma situación es completamente diferente para una persona que puede usar fácilmente las escaleras y que tal vez prefiera esa ruta en lugar de una más larga.

Como todos vemos el mundo de manera diferente, no existe una solución única que pueda satisfacer todas nuestras necesidades. Eso implica que al encontrar rutas adecuadas en un mapa, como las peatonales, es crucial entender quién es el usuario final. Las preferencias (su modelado y adecuada gestión) son la clave para definir lo que es importante para cada individuo.

El objetivo de este trabajo es el uso de técnicas de Soft Computing para la mejora y resolución de modelos de decisión y optimización en el escenario de mapas adaptativos. El objetivo se divide en tres subobjetivos: definir un modelo para resolver el problema de la ruta personalizada y proporcionar un enfoque de solución, definir un modelo para mapas adaptativos y proporcionar un enfoque de solución y definir un método a posteriori para seleccionar rutas incorporando preferencias después del proceso de optimización.

Para ese propósito, se presenta en detalle el problema de la ruta personalizada, se propone un modelo y se proporciona un enfoque de solución. Además, se presenta, discute y desarrolla el concepto de mapas adaptativos. En resumen, un mapa adaptativo es un mapa capaz de mostrar cómo un usuario ve el mundo no

solo en términos de distancias, sino también considerando otras características cuya importancia está determinada por ese mismo usuario.

La obtención de rutas personalizadas (en términos de las preferencias del usuario) también se puede alcanzar mediante un enfoque diferente y más general: la selección de soluciones de interés. Este método a posteriori consiste en seleccionar rutas personalizadas partiendo de un amplio conjunto de rutas, obtenidas por la resolución de un problema multiobjetivo, que posteriormente son evaluadas teniendo en cuenta las preferencias del usuario identificando así las soluciones que son más interesantes para dicho usuario. En este trabajo también se considera y se explora este enfoque más general mediante la definición del problema de identificación de soluciones de interés y proporcionando un método de resolución.

Chapter 1

Introduction

In this introduction the motivation for this work is first explained, together with the context in which the research is developed. The main objective and the sub-objectives for this work are detailed next. Finally, the structure for the thesis is commented and the publications related to the research are listed.

1.1 Motivation

Both the Spanish Strategy for Science, Technology and Innovation and the European Union's Framework Programme for Research and Innovation "Horizon 2020" have identified a set of challenges the society will have to face in the middle and long term, e.g., intelligent transport, health, safe societies and tourism. In each one of these challenges it is possible to recognise

decision and optimisation problems leading to models and frameworks of any nature (mathematical, linguistic, computational, etc.) requiring suitable solution methods and algorithms.

This thesis has been developed within a research training staff grant associated with the project TIN2014-55024-P and co-funded by the European Social Fund. This project raises as initial hypothesis the fact that decision and optimisation models can be more efficiently addressed by using Soft Computing techniques and, consequently, better solutions can be found for the corresponding social challenges.

Such improvements are articulated in two aspects. First, the formalisation and study of the models that will be used to approximate reality. Second, the resolution of real world problems that can be conceptualised by the aforementioned models. In both cases, it is hypothesised that (a) Soft Computing tools, particularly fuzzy sets and systems and meta-heuristics, are ideal to tackle such problems, and (b) a better understanding of the models and their solution methods will yield suitable solutions for those problems.

On this project three scenarios are identified and explored: the study of location models with coverage (typically applicable to situations of service deployment, and directly associated to situations of security assurance of those services), the development of the concept of "adaptive maps" (especially suited for route design problems where the routes depend on user features, and thus closely related to intelligent tourism schemes), and multi-criteria

decision models as an essential tool in decision making problems which are focused here on crisis and emergency management. The adaptive maps scenario is the context in which this work focuses.

The main motivation for the adaptive maps research is that the applications based on maps usually do not consider the uncertainty and imprecision of the real world characteristics. Also, each person could have a different view of the world having different priorities and difficulties. For example, a person with a disability could see in a map that his/her destination is only 20 meters away following a single street but, in the real world he/she can see stairs on the path that make that route of 20 metres non-viable. That same situation is completely different for a person that can easily use the stairs. The idea of adaptive maps is to provide maps that better represent the characteristics of the world according to each person's needs and preferences.

In this work, the personalisation is addressed in two different ways. First, the preferences are incorporated prior the optimisation problem and, secondly, they are incorporated after the optimisation process solving a decision making problem to identify the best solutions for the current preferences.

1.2 Objectives

The general objective for this work is the use of Soft Computing for the improvement and resolution of decision and optimisation models on the adaptive maps scenario. In such scenario two main problems are identified: the personalisation of the routes and the visualisation of the "adaptive" map.

Following the general objective, next sub-objectives are defined:

1. Define a model to solve the personalised route problem and provide a solution approach.
2. Define a model for adaptive maps and provide a solution approach.
3. Define a posteriori method to select routes by incorporating preferences after the optimisation process.

1.3 Structure

The thesis is structured into three parts, each one is related to one of the sub-objectives, in addition with a final chapter that details conclusions and future work. Resulting publications related to each one of the sub-objectives are listed next.

The definition of a model to solve the personalised route problem is address in Chapters 2 and 3. A solution approach is also proposed.

The personalised route problem (PRP) is first defined as finding a route that suit the user's preferences or needs. Then, a model based on incorporating the preferences prior the optimisation process is detailed (a priori approach). Finally, a solution approach is proposed and an implemented Android app, called PRoA (personal route assistant), is shown in detail. The results from the research on PRP are published in:

> Torres, M., Pelta, D. A., and Verdegay, J. L. (2018c). PRoA: An intelligent multi-criteria personalized route assistant. *Engineering Applications of Artificial Intelligence*, 72:162–169.

Also, PRoA app is awarded with the first place on an Artificial Intelligence's apps competition:

> Torres, M., Pelta, D. A., and Verdegay, J. L. (2016). PRoA: una Aplicacion Android para el Diseño de Rutas Personalizadas. In *Congresos de la Asociación Española para la Inteligencia Artificial, Salamanca.*

Chapter 3 continues the research on personalised routes. The presented work consists on modifying previous PRP model to add tolerance on the constraints. With this new model, based

on fuzzy constraints, the user's preferences are more relaxed and, potentially, more interesting solutions can be found. Two publications are related to PRP with fuzzy constraints:

> Torres, M., Pelta, D. A., Cruz, C., and Verdegay, J. L. (2017). Personalized route problem with fuzzy constraints. In *2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. IEEE.

> Torres, M., Pelta, D. A., and Lamata, M. T. (2018b). A new approach for solving personalized routing problems with fuzzy constraints. In *2018 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. IEEE.

The personalisation of routes is addressed before the adaptation of the maps. By obtaining routes than can suit the user's preferences it is easier to understand how to adapt a complete map.

The definition of a model for adaptive maps and a solution approach sub-objective is developed in Chapter 4. A model is proposed and a generation method is detailed. The results consist on map images that are deformed to represent the preferences of each user. The adaptive maps research is summarised in two publications:

Torres, M., Pelta, D. A., Verdegay, J. L., and Cruz, C. (2018e). Towards adaptive maps. *International Journal of Intelligent Systems*, 34(3):400–414.

Torres, M., Pelta, D. A., and Verdegay, J. L. (2018d). A proposal for adaptive maps. In *Communications in Computer and Information Science*, pages 657–666. Springer International Publishing.

Chapter 5 includes the research related to the definition of a posteriori method to select routes by incorporating preferences after the optimisation process. As the proposed method is suitable for multiple fields, this chapter is not only focused on adaptive maps and personalised routes, but in a more general way on how to analyse and filter a set of solutions according to the user's preferences. Next publication is related to this research:

Torres, M., Pelta, D. A., Lamata, M. T., and Yager R. R. (under review). An Approach to Identify Solutions of Interest from Multi and Many Objective Optimisation Problems. Submitted to Knowledge-Based Systems.

Also, a research stay in Newcastle (UK) in collaboration with Interdisciplinary Computing and Complex BioSystems (ICOS) research group, Newcastle University, allowed to apply the pro-

posed method on the synthetic biology field. The research is published as:

> Jiang, S., Torres, M., Pelta, D., Krabben, P., Daniel, R., Luzardo, J. T., Kaiser, M., and Krasnogor, N. (2018). Improving microbial strain design via multiobjective optimisation and decision making. In *AI for synthetic biology 2, IJCIA*, pages 1–6, Stockholm, Sweden.

A second publication is also obtained in collaboration with ICOS research group:

> Torres, M., Jiang, S., Pelta, D., Kaiser, M., and Krasnogor, N. (2018a). Strain design as multiobjective network interdiction problem: A preliminary approach. In *Advances in Artificial Intelligence*, pages 273–282. Springer International Publishing.

The conclusions are summarised in chapter 7 where possible extensions and improvements to this work are also detailed.

Chapter 2

Personalised route problem

The Personalised route problem (PRP) consists, in general terms, on finding routes that suit the user's preferences. The main contribution of this chapter is to provide a model that can be adapted into multiple scenes, situations or circumstances that could determine the user's decision and preferences. For that purpose, a PRP model able to represent different scenes is proposed. Then, an example of scenario is defined: the pedestrian scene, where the aim is to find walking routes. Finally, to illustrate its application, an Android application called PRoA is shown in detail.

2.1 Introduction

The Personalised route problem aims to find a route according to the user's preferences. It resembles the so-called multi-criteria Shortest Path Problem [57].

In order to indicate the user's preferences, the scene needs to be defined first. Such definition is made by detailing the set of criteria that may, at the end, affect the user's decision. For example, in a pedestrian scene, the steps can be considered a criterion while, in a driving scene, traffic lights criterion better suits the user's needs. In the literature, the PRP is usually modelled for a specific scene and it is solved according to the considered criteria and nature of the scene itself with independence to any other PRP.

Probably one of the most explored Personalised route problems nowadays are within the tourism scene. In special, the Tourist Trip Design Problem (TTDP) [93] has been adapted to take into account the users preferences and better solving different needs, e.g. day tour routes where the users indicate their willingness to visit each point of interest within a tourist attraction [105] or a TTDP where the users can indicate the places and food they like, as well as constraints on cost and time [22]. In [9] several applications on PRP offering personalised touristic routes are compared and reviewed.

The PRP is also studied for different modes of transport. For example, the method presented in [68] is based on drivers'

preferences like distance, time, toll charges or scenery value, while the authors in [13] focus on bicycle routes allowing the rider to decide how important are points of interest's quality, distance among them and landmark density.

Research on Personalised routes for pedestrians is motivated because walking is not only a mode of transport, but a green and healthy one. The Institute for Transportation and Development Policy (ITDP) [1], based in New York, states that:

> "For decades, traditional transport planning has focused on improving conditions for private automobiles at the expense of safe sidewalks and bike facilities. Yet, the majority of the world's people rely on cycling, walking, and other forms of human-powered transport... Increasing the use of bicycles and the ease of walking is one of the most affordable and practical ways to reduce CO2 emissions, while boosting access to economic opportunity for the poor."

These environmental benefits, including moderate-intensity physical activity like walking or cycling, had proved to provide substantial health benefits [69]. Indeed, in the *National Travel Survey: England 2014* [92], walking has been found to be the most frequent transportation mode used for very short distance trips: 76% of all trips under 1.6 km are walks. In terms of averages

[1] https://www.itdp.org/what-we-do/cycling-and-walking/

per person, there were 200 walking trips registered in 2014 for a total of 290 km and 18 minutes per walking trip. In that survey, one can also read that the transportation modes accounting for most trips in 2014 were by car, either as a driver or a passenger (64%), whereas walking accounted for a 22%.

A research on urban transport done by the Spanish Ministry of Development, "Movilia"[2], shows the transport usage in Spain in the matter of daily urban routes, longer than 5 minutes walking, with data obtain from 2006. The results show that, during week days, 45.6% of the routes were done by walking or cycling, a percentage that is increased to 46.9% in weekend days.

As the literature indicates, both the environmental variables and the adequate facilities for walking, such as accessibility, are related to physical activity [75, 67]. In [17] the author states that "*Walking to work was significantly related to the environment score*". In that research, the score is calculated based on 18 neighbourhood characteristics (e.g., existence of accessible walking routes, such as sidewalks and paths, or available facilities like parks). In this way, a "good environment" or an "appropriate route characteristics" are significantly related to the decision of walking for example to work or just for leisure. However, those concepts are user-dependent and may vary according to each user's preferences.

[2]https://www.fomento.gob.es/informacion-para-el-ciudadano/informacion-estadistica/movilidad/encuesta-de-movilidad-de-las-personas-

Walking is also crucial for senior citizens or persons with a disability who may found many impediments on their routes such as stairs or steep slopes. Consequently, they need to conscientiously choose their route according to those constraining characteristics. Similarly, people with some mobility reduction, pregnant women, or people that walk with children may also be affected by the same constraints. Authors in [42] propose a routing method where wheelchair users can indicate their preferences on slope, steps, sidewalk condition and sidewalk traffic to obtain routes with less dangers and difficulties based on their needs. At the same time, authors in [10] study the influence of up to twenty-three physical characteristics related to urban environmental factors on the walking behaviour of elderly people, with he intention to find the right criteria to consider in that specific scene.

It is cleat that there is a relationship between the characteristics of a route and the user's decision to walk. This implies that a proper design of a route based on user's preferences is essential to promote walking activities. Some existing approaches are outlined next. A method to find the faster walking routes in open field where the criteria were defined according to the surface's friction is proposed in [3]. Similarly, [36] uses criteria such as waiting time, turns, or traffic lights to find a multi-modal route that includes walking and public transportation. The methods presented in [70] and [54] focused on the walkability concept or "pleasant" routes. Although these approaches are sound, they

generally need feedback from users to determine the street's walkability. This is the case of [95], which despite using some geographical data from different open data sources, currently, it is only working on four cities (Toronto, San Francisco, London and New York). Another case of dependence on feedback from a user is Vadeo [62], an app deployed in the city of Valencia, Spain, where the users have to indicate obstacles or impediments for persons with a disability. Once again, previous project cannot be considered enough to achieve a trustful system because it also depends on creating a new active community, as well as on the cooperation of its members. A different approach that does not depend on creating a new community of users is presented with Walkability Explorer [8]. The system uses data from OpenStreetMap to determine the walkability score. Previously, [48] used data from GIS to objectively measure features related to the walkability concept.

Given that diversity on scenes and their difference on considered criteria, a PRP model is proposed in a manner that it can be used for dealing not only with the pedestrian scene but with multiple scenes.

2.2 Personalised route problem

The PRP aims to find a route that suits the user's preferences. In this proposal the preferences are assumed to be known at the

beginning, that means, the users defines their preferences first, and then, the routing algorithm takes them into account to find suitable routes.

The preferences will be included in two different ways: as a weight measuring the importance to minimise or maximise each criterion, and/or as a constraint indicating a top boundary allowed for a specific criterion along the route, e.g., a 5% of maximum slope.

2.2.1 Model formulation

A model for the PRP is formulated next.

The PRP is described with a finite number of criteria and constraints. The objective is to obtain a route from a starting to an ending points according to the user's preferences on a set of criteria. It assumes a directed and finite graph $G = \{N, A\}$ where $N = \{1, ..., n\}$ is a finite set of nodes and $A \subseteq N \times N$ is a finite set of directed arcs denoted by (i, j) where $i \neq j$ and $i, j \in N$. The arcs in A represent street segments from a map.
Then, denoting

G a finite graph $G = \{N, A\}$,

i, N indexes and set of nodes of the graph G,

A set of arcs of the graph G denoted as (i, j),

s, t the starting and ending nodes, $s, t \in N$,

k, F indexes and set of criteria,

k, W index and set of weights that represent the importance of minimising each criterion F_k,

k, M index and set of the maximum values allowed for each criterion F_k on every arc in A,

A' the subset of arcs that satisfy the constraints, $A' \subseteq A$,

x decision variable vector with $x_{ij} = 1$ if the arc (i, j) is part of the solution path, $x_{ij} = 0$ otherwise,

the model is formulated as follows:

$$\min \quad \sum_{(i,j)\in A'} \sum_{k=1}^{|F|} x_{ij} c_{ij}^k w_k, \tag{2.1}$$

s.t.

$$\sum_j x_{ij} - \sum_j x_{ji} = \begin{cases} 1 & \text{if } i = s, \\ 0 & \text{if } i \neq s, t, \\ -1 & \text{if } i = t, \end{cases} \tag{2.2}$$

$$A' = \{(i, j) \in A | c_{ij}^k \leq M_k \quad \forall k\}, \tag{2.3}$$

$$\sum_{k=1}^{|F|} w_k = 1 \quad \textit{and} \quad 0 \leq w_k \leq 1 \quad \forall k, \tag{2.4}$$

$$x_{ij} \in \{0, 1\} \quad \forall (i, j) \in A', \tag{2.5}$$

where c_{ij}^k is the cost for the arc (i, j) on the criterion $k \in F$, a normalised value that will be minimised or restricted. The objec-

tive function 2.1 minimises the total cost of the path. Constraint 2.2 ensures that the arcs of the path are connected by a node to the next arc and the path starts on s node and ends on t node. Constraint 2.3 defines the subset $A' \subseteq A$ with the arcs that satisfy the constraints. Constraint 2.4 is a normalisation condition for weights.

Depending on the nature of a criterion k, its cost c_{ij}^k could be a numerical value, e.g., elevation, latitude, longitude or length, or a nominal characteristic like the edge's street type, e.g., path, ban or pedestrian.The criteria and their costs' will depend on the scene that is been solved. In other words, in order to solve the PRP on a specific scene, both the criteria and their costs have to be previously described and calculated.

Given a specific scene, the model only requires the user to indicate the weights W_k and/or the maximum values allowed M_k for each criterion $k \in F$.

2.3 Example: pedestrian personalised route problem

The rest of the chapter focuses on the scene of pedestrian routes as an application of the model proposed in previous section. That is, the PRP is solved within the pedestrian scope taking into account the diversity on users. For example, an elderly person could desire to have a walk with no stairs, and a maximum up-

ward and downward slopes of 10% while enjoying parks instead of high traffic streets. An other example could be a family that want to go to the city centre going throw pedestrian streets and reducing both the length of the final path and its slopes.

The PRP in a pedestrian scene is solved by specifying the criteria that will be taken into account for the pedestrians and then defining the costs' calculation. The pedestrian scene used on this work is defined by criteria like distance, slopes, pedestrian adequacy of the area (criterion called green zones) and stairs. More specifically the set is defined as $F = \{distance, upward\,slope, downward\,slope, green\,zones, stairs\}$. Those criteria are used in the PRP formulation in two different ways: optimisation criteria and constraints. The optimisation criteria will be part of the objective function in the PRP model (see Eq. 2.1), while the constraints indicate if a specific arc should be included or not in A' set (see Eq. 2.3), either because it exceeds some limit, e.g., *slopes*, or contains a possibly undesirable criterion, e.g., *stairs*.

As it has been previously stated, in order to solve the PRP in a pedestrian scene, the costs $c_{ij}^1, ...c_{ij}^k, ...c_{ij}^{|F|}$ need to be calculated first. In order to calculate the costs according to the selected criteria ($F = \{distance, upward\,slope, downward\,slope, green\,zones, stairs\}$), information about the nodes and arc in the graph is needed. In the defined scene, the required data are arc's length, node's elevation and type of street associated to the arc.

The costs' calculations for the PRP in the selected pedestrian scene are described and discussed next. It should be clear that

there is not an unique way to calculate them and the chosen method will have a huge impact on the solutions obtained.

Distance criterion

The cost associated to the *distance* criterion, F_1, is calculated as:

$$c_{ij}^1 = l_{ij}/d_{max}, \tag{2.6}$$

where l_{ij} is the length in meters of the arc $(i,j) \in A$ and $d_{max} = max\{l_{ij} \quad \forall (i,j) \in A\}$ is the maximum length for the arcs in A.

Slopes criteria

The $upward\,slope$ and $downward\,slope$ criteria, F_2 and F_3, could be calculated following different formulations. For example, a strait forward way to obtain the costs could be

$$c_{ij} = |e_i - e_j|, \tag{2.7}$$

where e_i is the elevation value in meters for the node $i \in N$. This value only represents the variation on the elevation without taking into account the distance between the nodes or the real slope. The costs could be more precise and represent the slope for $upward\,slope$ and $downward\,slope$ respectively as:

$$c_{ij}^2 = \begin{cases} \frac{|e_i - e_j|}{l_{ij}} & \text{if } e_i < e_j, \text{ and } l_{ij} \geq 1, \\ 0 & \text{otherwise,} \end{cases} \tag{2.8}$$

$$c_{ij}^3 = \begin{cases} \frac{|e_i - e_j|}{l_{ij}} & \text{if } e_i > e_j, \text{ and } l_{ij} \geq 1, \\ 0 & \text{otherwise.} \end{cases} \tag{2.9}$$

The condition $l_{ij} \geq 1$ avoid possible problems caused by an extremely short arc in the graph (less than one meter) because it could cause distortions on the calculations.

However, this formulation will not assign different costs based on the slope's length. For example, an arc of 100 meters with 10% of slope will have the same cost as an arc of 1 meter with 10% of slope. Despite this method could be useful in certain scenes, in the pedestrian scene considered for this work, the *upward slope* and *downward slope* criteria' costs are obtained as follows:

$$c_{ij}^2 = \begin{cases} \frac{|e_i - e_j|^2}{l_{ij} d_{max}} & \text{if } e_i < e_j, \text{ and } l_{ij} \geq 1, \\ 0 & \text{otherwise,} \end{cases} \tag{2.10}$$

$$c_{ij}^3 = \begin{cases} \frac{|e_i - e_j|^2}{l_{ij} d_{max}} & \text{if } e_i > e_j, \text{ and } l_{ij} \geq 1, \\ 0 & \text{otherwise,} \end{cases} \tag{2.11}$$

where e_i is the elevation value in meters for the node $i \in N$ and l_{ij} is the length of the arc $(i, j) \in A$. This formulation is chosen

on current scene because it punishes longer slopes. That means, that the longer arcs, with same slope percents, will have greater costs.

This formulation is not intuitive for an user that has to decide the maximum values allowed for slopes. The formula in Eq. 2.8 is more intuitive, so it is a better option to define the constraints for slopes criteria. To address this issue, the costs c^2_{ij} and c^3_{ij} calculated by equations 2.10 and 2.11 are used in the objective function, while the constraint uses costs c'^2_{ij} and c'^3_{ij} calculated by Eq. 2.8, a more intuitive value to facilitate the user's understanding of the constraint.

Green zones criterion

The cost associated to the *green zones* criterion, F_4, is calculated as:

$$c^4_{ij} = greenCost_{ij} l_{ij} / d_{max}, \tag{2.12}$$

where $greenCost_{ij}$ indicates how bad is the arc (i, j) in terms of "green zones" and it is defined in a simple way according to the type of the streets as:

$$greenCost_{ij} = \begin{cases} 0.001 & \text{if track,} \\ 0.005 & \text{if path,} \\ 0.01 & \text{if stairs or pedestrian,} \\ 1 & \text{otherwise,} \end{cases} \tag{2.13}$$

where tracks are referred to unpaved roads like forest tracks; paths are trails open to all non-motorised vehicles, e.g., hiking trails or bike trails; and pedestrian streets are auto-free zones and car-free zones. It should be understood that those values are empirically selected to punish the streets that are not a "green zone" favouring the tracks and paths which are usually found in fields and parks.

Stairs criterion

Finally, the cost associated to the *stairs* criterion, F_5 is calculated as:

$$c_{ij}^{5} = \begin{cases} 1 & \text{if stairs,} \\ 0 & \text{otherwise.} \end{cases} \tag{2.14}$$

To solve a PRP the user only has to indicate the starting and ending nodes, s and t and the preferences which are included

in the model as weights, W_k, and maximum values allowed, M_k, for each criterion $k \in F$. It is clear that multiple profiles can be represented by setting different weights, W_k, and/or maximum values, M_k. In the pedestrian scene, the approach is the same one. For example, an elderly looking for a walk with no stairs and a maximum upward and downward slopes of 0.1 and parks instead of high traffic streets could set $W_k = \{ 0, 0, 0, 1, 0\}$ (all the weight to the *green zones* criterion) and $M_k = \{ 0, 0.1, 0.1, 0, 1\}$ (with maximum values for *upward slope*, *downward slope* and *stairs* criteria). Similarly, the family that want to go to the city centre using pedestrian streets and reducing both the length of the final path and its slopes could indicate $W_k = \{ 0.2, 0.2, 0.2, 0.4, 0\}$ (or any other combination of weights for *distance*, *upward slope*, *downward slope* and *green zones*) and $M_k = \{ 0, 0, 0, 0, 0\}$ (no constraints needed). See Table 2.1 for more profile examples.

The model allows the duplicity of a criterion in case it does not have same behaviour in the optimisation and the constraints parts. For example, the slopes could be calculated considering the arc distance to punish longer slopes in the optimisation process while being considered as simple slope's percentage on the constraints. Two criteria are need in that case: *optimisation slope* and *constraint slope*. Same procedure can be done with the rest of criteria if the modelled scene requires it.

2.4 Pedestrian personalised route problem implementation

PRoA (Personalised Route Assistant) [84] is an Android app that solves the PRP within the pedestrian scene explained before.

PRoA is targeted to Android 5.0 devices and it is built using third party services: OpenStreetMap Server Side Scripting to obtain the streets map data, mapQuest Open Elevation Service to request the elevation information, Google APIs (Maps Android, Directions, Elevation, Places), AndroidPlot to build graphs showing the route's elevation and SQLite as a database manager.

Walk-through

The interface and work-flow have been designed to fit all information on a smart-phone screen as well as on a tablet screen and making it easy to interact with the application. The functionality is divided into three tabs that corresponds to the steps the user will face in order to obtain the route: the plan input data, the importance criteria specification and the routes visualisation.

In the first tab, the user indicates the origin and destination points, s and t, by placing markers on a map or writing the address on a search field that offers "auto-complete" functionality and suggestions. This tab is shown in Fig. 2.1 (a).

The criteria importance, presented on the second tab, allows the user to indicate the preferred importance for each criterion.

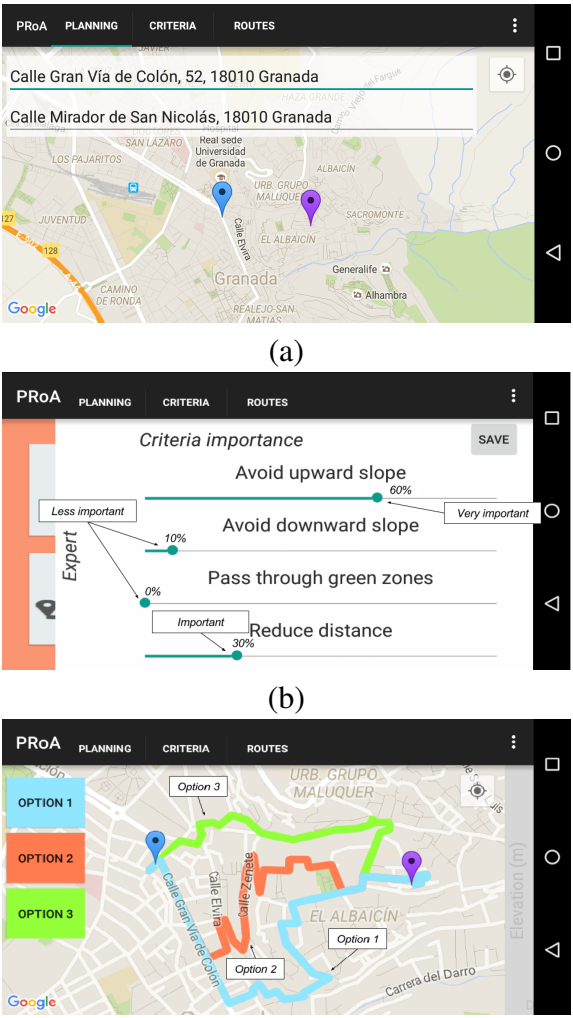
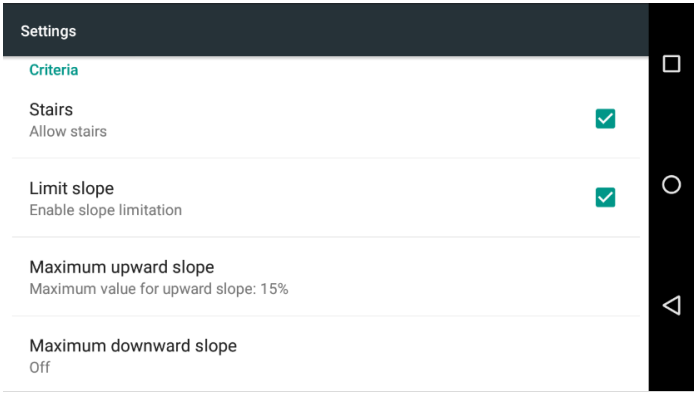
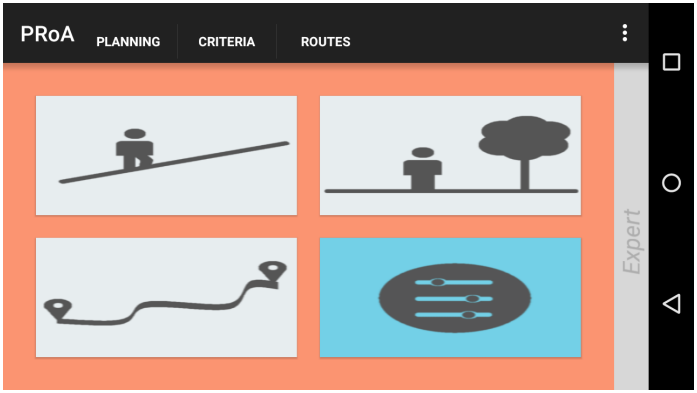


Fig. 2.1 PRoA app steps: (a) plan input data tab, (b) user defined criteria importance and (c) the routes found with the criteria specified in (b).



(a)



(b)

Fig. 2.2 PRoA app screens: (a) “settings” options for avoid stairs and slope limitation and (b) available pre-defined profiles screen.

The user can make a fully personalised specification of his/her interests as shown in Fig. 2.1 (b).

Figure 2.1 (c) shows the alternatives provided by PRoA for the criteria importance distribution shown in Fig. 2.1 (b). Those routes are shown in the third tab of the application. The calculated route and alternatives depicted on the map are identified by colours. The user can select and deselect each route in order to show or hide its path and elevation data (provided line chart). When an alternative is selected, the user can request the directions to see the list of instructions on how to follow the route. The routes can also be saved and loaded later. This criterion allows an easy comparison between the routes obtained by different criteria preferences.

The slopes limitation and the stairs avoidance option are indicated in application's settings, as illustrates Fig. 2.2 (a), and can be modified as needed. Also, three predefined profiles are available for the user to select, as shown in Fig. 2.2 (b). The predefined profiles are called "avoid slopes", "reduce distance" and "pass through green zones" and their corresponding weights values are summarised in Table 2.1.

PRoA system

PRoA has three main components: the Data Manager, the Criteria Evaluator and the Route Builder, each one associated with a different stage.

Profile	W_1	W_2	W_3	W_4	W_5
Avoid slopes	0.1	0.45	0.45	0.0	0.0
Reduce distance	1.0	0.0	0.0	0.0	0.0
Pass through green zones	0.3	0.0	0.0	0.7	0.0

Table 2.1 Weights for the predefined profiles being $F = \{distance, upward\,slope, downward\,slope, green\,zones, stairs\}$ the set of criteria.

A PRoA session lead to the work-flow shown in Fig. 2.3. First, the user specifies the origin and destination locations, s and t, and the preferences and constraints, W and M. Then, the Data Manager, using origin and destination, constructs the needed street graph, G. This component obtains the required data related to the criteria F from a local database. In case there is some missing information, it requests the absent information using web services and stores it in the device. Once the street graph G is completed, the Criteria Evaluator calculates the costs c_{ij}^k $\forall(i,j) \in A$, $k \in F$ and creates the set of available arcs A'. After that, the Route Builder finds the desired route (from origin to destination) and up to two alternatives to the first one.

Data Manager

The Data Manager component obtains data from OpenStreetMap (OSM) [64], a project that creates and distributes open and free geographical data. The OSM data set has been compared to similar data sets and results have proved its good data quality

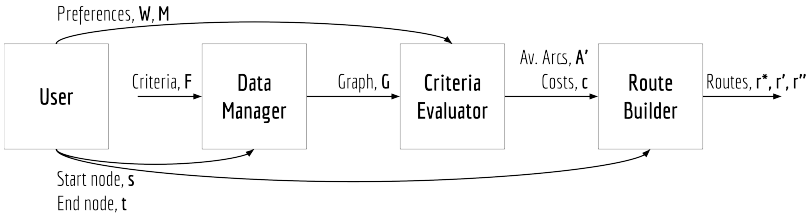


Fig. 2.3 Work-flow diagram for a PRoA session. The user indicates preferences and start and end nodes. The Data Manager build the graph using criteria information and the selected nodes. The Criteria Evaluator component uses the preferences and the graph to calculate the costs and the available arcs which are later used by the Route Builder component. The result consists on three routes.

[33]. Also, [108] concluded that routes planned with OSM data "*resulted in shorter shortest paths for pedestrians than commercial data sets*" in a study of the pedestrian routes on German and U.S. cities.

The data requested to OSM is:

- *Nodes*: the node is a point defined by its latitude, longitude and id.
- *Streets*: streets in OSM are an ordered list of nodes. Note that a node from a street could be an intersection of streets or a middle point of the street that connects two segments with different orientation.

- *Street type*: if it is pedestrian, road, path, track, steps, etc.

The elevation data is not provided by OSM, instead it is requested to other services based on OSM datasets that also include the elevation map.

Based on the previously described information, the Data Manager component calculates other important attributes:

- *Edges*: an edge in the graph is a segment of a street in OSM, that is, a connection between two nodes. A node in OSM is not always a street intersection so an edge of the graph can be connected to one edge (same street continues) or more edges (intersection).

- *Distance*: the length (in meters) of an edge is calculated as the distance between its connecting nodes.

- *Elevation variation*: the elevation variation (in meters) of each edge is calculated as the difference on the elevation data of its two nodes.

The system is designed with a local database that allows an offline use of the app. The database is updated each time the user first requests a route on a specified area of the map. For this purpose, the map's data is split into tiles. A node from the graph belongs to one and only one tile making easier to index the database. This method ensures data consistency because the

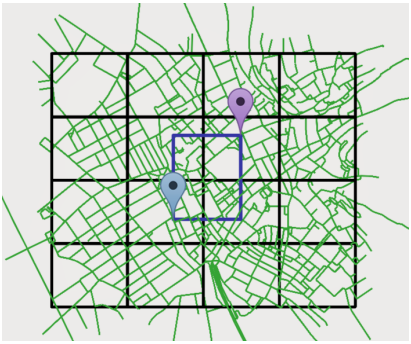


Fig. 2.4 Four tiles cover the Bounding Box area defined by the two markers. Also shown 12 adjacent tiles that are needed to construct the street graph.

tile is always fully requested and stored on the database so no data is missing if the tile is already stored.

The tiles requested to OSM are all the tiles that contains any information inside the Bounding Box (an OSM concept described as the area defined by two longitudes and two latitudes) formed by the origin and destination points and, also, all tiles adjacent to that Bounding Box. This approach limits the data available to build the route at the Route Builder component, and because of this limitation the tiles needs to be of a considerable size. After some empirical testing, the defined size of the tiles is 0.01 degrees variation on latitude and 0.01 degrees variation on longitude. To illustrate the Bounding Box concept and the tiles requested for a specified plan route, Fig. 2.4 shows an example of the tiles needed. In the example the tiles are smaller to facilitate

the visualisation: 0.002 degrees variation on latitude and 0.003 degrees variation on longitude.

Criteria Evaluator

The Criteria Evaluator component is in charge of all costs' calculation $c_{ij}^{k}\ \forall(i,j) \in A, k \in F$. It also creates the set of available arcs A' with the ones that satisfy the constraints. For those purposes this components uses the user's preferences information: the weights, to calculate the costs, and the maximum values, to validate the constraints and create A' set.

Route Builder

The proposed routes calculation is based on the A* algorithm shown in Alg. 1. The algorithm requires as input the origin (s) and destination (t) nodes and two functions (shown underlined) $cost(i,j)$ and $h(i)$. The former determines the cost c_{ij} of the arc (i,j) connecting nodes i and j. The function $h(i)$, shown in Alg. 2, calculates an admissible heuristic estimation from node i to t, the end node. The estimation depends on the *distance*, *downward slope* and *upward slope* criteria, while currently, there is no estimation value for the criterion *green zone*.

A single run of the A* algorithm finds a solution. In order to obtain alternatives routes, PRoA uses an approach that consist on re running the same A* algorithm but previously banning a set of nodes. More specifically, given the previous route r^*,

on a second run of the algorithm, the *ClosedSet* is initialised with those nodes from r^* belonging to the second third of the route. This way, the A* algorithm does not expand or evaluate those *banned* nodes. As a result, a new route r' will be obtained, if possible. To obtain the third route r'', the *ClosedSet* will be initialised with the nodes in the second third part of the routes from both r^* and r'.

This approach can be used as many times as alternatives desired. However, it should be clear that it is not always possible to obtain a solution. For example, if two nodes connecting a street, which is the only way to connect two separate areas, are banned, the method will not find an alternative. The nodes banning method in the second third part of the route is arbitrary but produces good results in practice. Other alternatives can be naturally explored.

2.4.1 Analysis and results in Granada, Spain

The assessment of PRoA is performed through a set of real world examples developed in Granada, Spain. Three analysis are presented. The first one compares the routes provided by PRoA with those provided by "de-facto" routing application Google Maps. Then, in the second part, two examples with slopes limitation, green zones and stairs avoidance are shown. In this case, comparisons against other tools are not possible as such features (to the best of author's knowledge) are not considered

Origin		Destination		Distance		
Latitude	Longitude	Latitude	Longitude	PRoA	GM	Reduction
37.178013	-3.595665	37.183169	-3.599816	777	816	4.8%
37.178346	-3.596976	37.178904	-3.596161	142	230	38.3%
37.179344	-3.597739	37.178037	-3.593710	517	550	6.0%
37.175637	-3.594760	37.178022	-3.593719	444	500	11.2%
37.177094	-3.595219	37.179450	-3.594187	348	450	22.7%
37.182653	-3.598963	37.183898	-3.597441	281	350	19.7%

Table 2.2 Illustrative examples of shorter distances routes found by PRoA against Google Maps (GM).

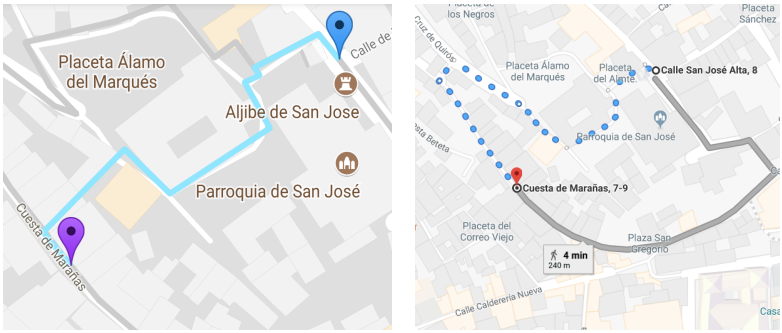
by any other available application. Finally, the third part shows a brief performance analysis.

Route lengths comparison

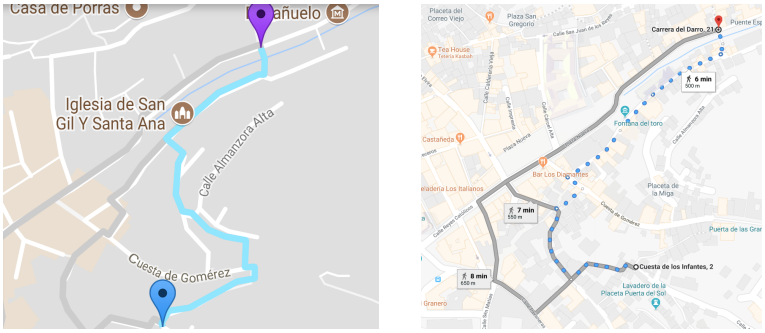
Table 2.2 shows the origins and destinations of six routes, together with their distances according to PRoA and Google Maps. All the routes are located in Granada (Spain). PRoA was ran with the "reduce distance" profile for a fair comparison.

As it can be observed, PRoA routes are shorter than those provided by Google Maps. The reduction in length goes from almost 5% to an impressive 38% in the shortest route (142 m vs. 230 m). This is a crucial point from a pedestrian point of view. The routes for two of those examples are shown in Fig. 2.5.

The reason underlying these differences is the OSM data completeness for pedestrian routes. This fact was already known [33], but it can be confirmed in the case of Granada where a lot



(a)



(b)

Fig. 2.5 Examples of different routes obtained by PRoA (left) and Google (right). On (a), case number 2 from Table 2.2 with a reduction of 38.3%. On (b), case number 4 from Table 2 with a reduction of 11.2%.

of small streets, alleyways and passages are properly mapped in OSM, but not in the Google proprietary maps.

Beyond distance minimisation

In this part, the role of "personalisation" is emphasised by showing how different routes can be obtained for different user preferences.

Example 1 (Different User Profiles). Their preferences correspond with the following sets of weights W shown in Fig. 2.6: (a) $\{W_1 = 1\}$ for user 1, (b) $\{W_2 = 1\}$ for user 2 and (c) $\{W_1 = 0.5, W_2 = 0.5\}$ for user 3. Fig. 2.7 (a) shows the routes obtained according to every profile for users 1, 2 and 3 respectively, and Fig. 2.7 (b) allows the user to analyse the elevation profile. It can be observed how the route corresponding to option 2 allows to reach the destination using a larger but "softer" (in terms of elevation) path.

Example 2 (Different Constraints). This second example, shown in Fig. 2.8 and Fig. 2.9, illustrates the differences on the routes obtained when setting slope limitations and not allowing stairs. The routes go from $(37.184332, -3.602442)$ to $(37.181191, -3.592776)$. First, routes on Case 1 are obtained under the "reduce distance" profile with no other constraints. Cases 2 and 3 are both obtained with the "pass through green zones" profiles. Case 2 has a constraint on maximum slope of 30% and allows stairs while routes from Case 3 have the same

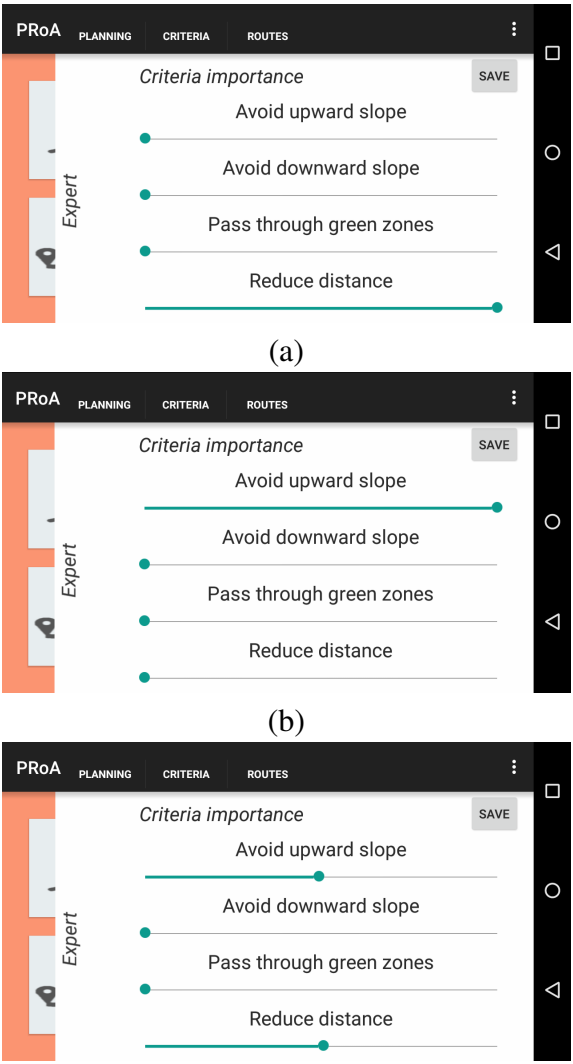
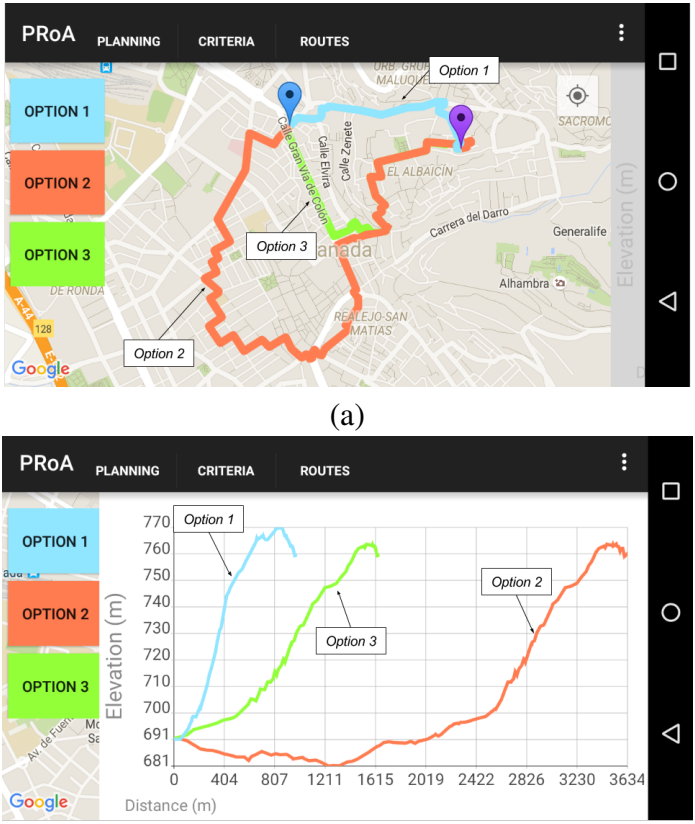


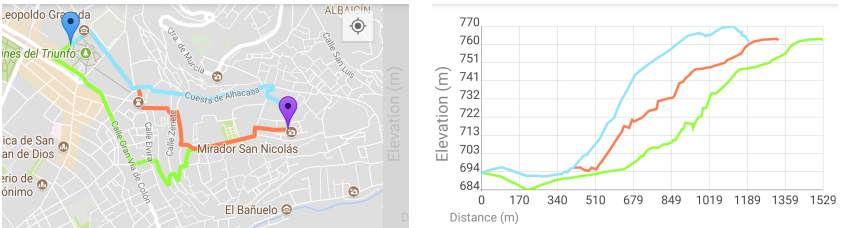
Fig. 2.6 Three different user defined profiles are shown in (a), only reduce distance, (b) only reduce upward slopes, and (c), reduce both upward slopes and distance.



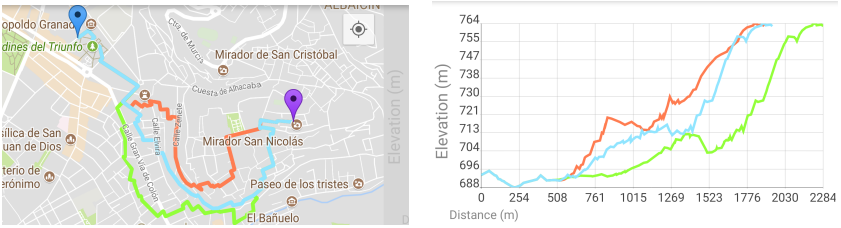
(a)

(b)

Fig. 2.7 Routes obtained in (a) and their elevation profiles shown in (b).

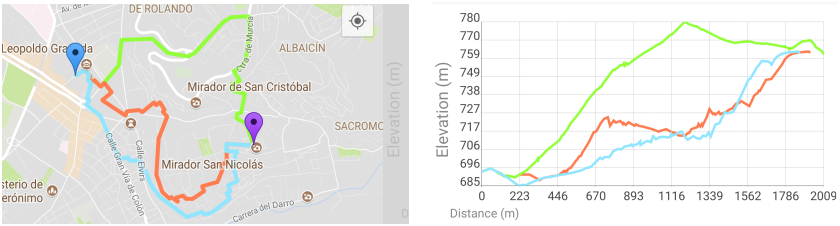


case 1

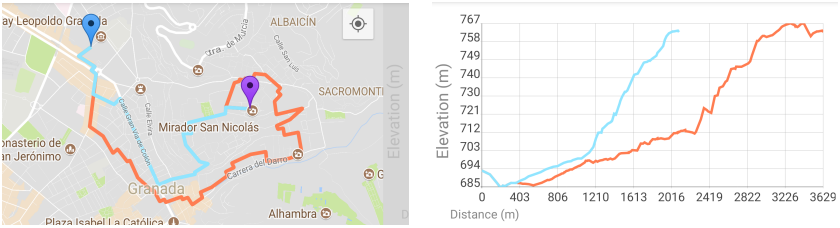


case 2

Fig. 2.8 Use of limitations. Case 1: “reduce distance” profile and no limitations. Case 2: “pass through green zones” profile with a limitation of 30% on maximum slopes and stairs allowed; The corresponding elevation profiles are also shown.



case 3



case 4

Fig. 2.9 Use of limitations. Case 3: “pass through green zones” profile with a limitation of 30% on maximum slopes and stairs not allowed; and Case 4 “avoid slopes” profile with a limitation of 25% on maximum slopes and stairs not allowed. The corresponding elevation profiles are also shown.

slope constraint but do not allow stairs. Finally, Case 4 uses the "avoid slopes" profile with a constraint on maximum slope of 25%, also not allowing stairs. All examples clearly demonstrate that PRoA properly solves the Personalised route problem oriented to pedestrians. A wide variety of routes can be obtained as different user profiles and requirements are defined.

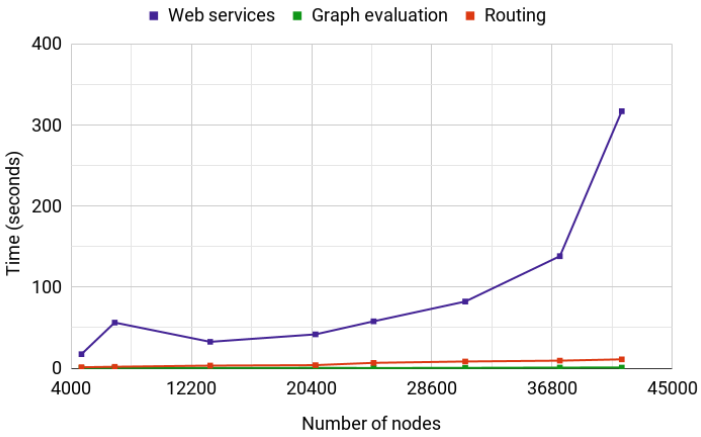
Performance analysis

A brief performance analysis is next detailed to provide information regarding the running time of the application. Eight routes are defined in Granada, that lead to different underlying graphs ranging from 4000 to 42000 nodes. PRoA is run with the "reduce distance" profile, measuring the computational time of every computational stage.

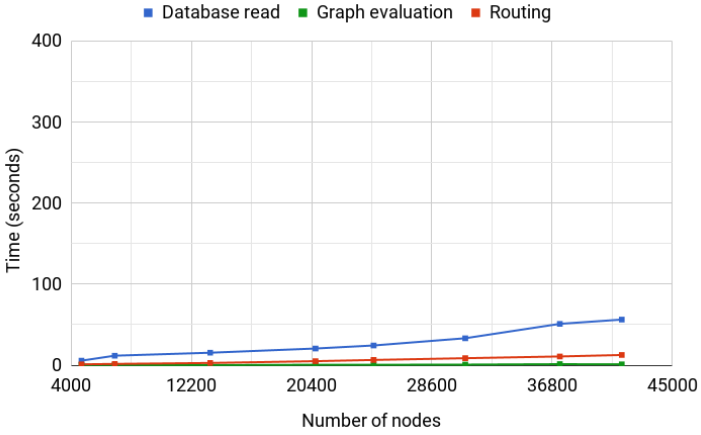
Two situations are considered for the analysis: when the map information (the database) is not available in the mobile phone and when it is. Fig. 2.10 (a) shows the former situation while (b), the latter.

It is clear that the information request from Web Services is the most time consuming part. The times displayed should be considered just as an illustrative examples because a high variability should be expected due to different response time from OSM servers throughout the day.

When the map is already available in the device, the time needed to manage the stored information is clearly reduced. In



(a)



(b)

Fig. 2.10 Running times versus map size (number of nodes on the graph) (a) requiring the download of the database and (b) when the database is already available.

the biggest case considered, with a graph with 41563 nodes, the map processing stage took less than a minute.

Regarding the computational times of the graph evaluation and the routing stages, it can be observed a linear behaviour of the algorithms with a very small slope, indicating good scalability and speed.

2.5 Conclusions

A "good environment" or "appropriate" route characteristics are concepts significantly related to the decision of walking instead of using any other mode of transport, for example, when going to work or just for leisure. However, these concepts are user dependent and may vary according to each one's preferences. So, towards emphasising the role of walking as transportation, there is a clear need for route personalisation based on the user's preferences and limitations, that is, to solve the personalised route problem (PRP).

In this chapter, the PRP has been defined and its model has been described. The versatility of the model is noticeable. Different PRP variations can be solved departing form this general model by defining a different set of criteria and/or constraints and, then, obtaining a different sets of costs for the arcs on the graph. With this approach, the PRP can be adapted not only to different scenes, e.g, bicycles routes, pedestrians routes, routes

for the elderly, routes for persons with a disability, touristic routes or safe routes for children going to school, just by defining the required criteria, e.g., bike lanes, crosswalks, traffic density or tactile ground surface indicators. Also, the model allows to incorporate more criteria into the problem just by defining their cost's calculation.

The solution for the PRP within a pedestrian scene is implemented as a full application named PRoA. The functionality, design and general implementation details of the app are shown and explained. Also, several examples located on Granada city are used to show a comparison of the routes obtained with different user's profiles.

From the analysis developed in the city of Granada, Spain, several conclusions can be drawn. In first place, PRoA routes can be shorter than those provided by Google Maps. This is due to the higher quality of OpenStreetMaps from which PRoA extracts the map information. In second place, PRoA is able to offer routes considering characteristics that other routing applications do not even consider. We provide a set of examples showing how the use of slopes limitation, green zones and stairs avoidance effectively led to different routes, which is of utmost importance for users with different preferences. Finally, and in terms of performance, it is confirmed that the running time increased in a linear way (with a very small constant) as the number of nodes in the underlying graph increased.

Algorithm 1 A* Algorithm.

function A*(s, t)
  $OpenSet \leftarrow \{s\}$
  $ClosedSet \leftarrow \{\}$
  $gCost[]$ initialized with default value of infinity
  $gCost[s] \leftarrow 0$
  $hCost[s] \leftarrow gCost[s] + \underline{h(s)}$
  while $OpenSet$ is not empty **do**
    $current \leftarrow node$ in $OpenSet$ with lower $hCost$ value
    if $current = t$ **then**
      return path($current$)
    end if
    for each $neighbor$ of $current$ **do**
      if $neighbor$ is not in $ClosedSet$ **then**
        $newGCost \leftarrow gCost[current] + \underline{cost(current, neighbor)}$
        if $neighbor$ not in $OpenSet$ **then**
          $OpenSet \leftarrow OpenSet \cup \{neighbor\}$
        end if
        if $newGCost < gCost[neighbor]$ **then**
          $gCost[neighbor] \leftarrow newGCost$
          $hCost[neighbor] \leftarrow gCost[neighbor] + h(neighbor)$
          $cameFrom[neighbor] \leftarrow current$
        end if
      end if
    end for
    $OpenSet \leftarrow OpenSet \setminus \{current\}$
    $ClosedSet \leftarrow ClosedSet \cup \{current\}$
  end while
  return no path found
end function

Algorithm 2 Cost estimation to go from a *node* to t, the end node.

function $h(node)$
  if $node = t$ **then**
    return 0
  end if
  $dist \leftarrow euclideanDistance(node, t)$
  $hDistance \leftarrow dist/d_{max}$
  $slope \leftarrow (elevation_t - elevation_{node})/dist$
  if $dist < 1$ **then**
    $hSlope \leftarrow 0$
  else
    $hSlope \leftarrow slope^2/d_{max}$ ▷ $hSlope < 0$ if downward slope
  end if
  $h \leftarrow hDistance W_1 + max\{hSlope, 0\}W_2 - min\{hSlope, 0\}W_3$
  return h
end function

Chapter 3

Personalised route problem with fuzzy constraints

The PRP has been solved including constraints on the criteria, e.g, maximum slopes limitation on the arcs. Those limitations are formulated as crisp constraints, that is, the value must be less than a desired value, i.e. “slopes lower than 5%”. However, for a typical user, it will be easier to indicate such constraints on the values using natural language which, in turn, may include vagueness, e.g., “slopes not much greater than 5%”. Also, an user could accept slopes of 5.1% on the solution path. This relaxation of the constraints can be achieved by considering some tolerance in the constraint’s accomplishment. For that purpose, those constraints should not be considered precise (crisp constraints) but remain fuzzy. With this example in mind and taking into

account the nature of the PRP it is clear that fuzzy constraints are useful not only for the pedestrian scene but for the PRP in general.

The purpose of this chapter is to adapt the PRP model to fuzzy constraints and explore its resolution following a parametric approach consisting on create α-cuts of the fuzzy problem. The alpha value used to generate the cuts will have a huge impact on the routes obtained as solution. A method to select the α values as an attempt to try to obtain a better set of routes is proposed and its behaviour is shown in an example.

3.1 Related work

The Shortest Path Problem (SPP) is a common and well studied problem but, unfortunately, little has been done with its fuzzy counterpart. The Fuzzy Shortest Path Problem (FSPP) [24] is usually based on fuzzy arc lengths, e.g., the multi-criteria shortest path in [63] where the arc length is considered as a fuzzy number. A similar approach is presented in [78], where a multi-objective path planning for construction sites is proposed using fuzzy numbers on different measures like safety and visibility. The method proposed in [35] also deals with negative parameters on the fuzzy arc lengths and offers a ranking on the solution paths. More recently, fuzzy numbers are used to represent the arc length on multiple metrics [23] while in [61] the Intuitionistic

Fuzzy Numbers are incorporated to deal with the FSPP. Later, fuzzy arc lengths are calculated according to several fuzzy numbers that represent attributes of the nodes [104]. More recently, costs related to arcs in the FSPP are represented using different and diverse techniques: interval-valued Pythagorean fuzzy numbers [25], triangular fuzzy number [81] or interval type-2 fuzzy numbers [21].

Despite FSPP has been solved with a variety of methods (e.g., a heuristic procedure addressed by [14], a hybrid genetic algorithm [38], a multiple labelling based algorithm [63], a modified intuitionistic fuzzy Dijkstra's algorithm proposed by [61] and a fuzzy Physarum algorithm presented in [104]), in FSPP, when constraints are included in the model they are usually crisp even when the arc lengths are fuzzy, see, for example, the routing algorithm for traffic engineering proposed in [45].

In order to solve the PRP with fuzzy constraints the so called parametric approach proposed in [94] could be applied. It has been recently used in many other applications and related problems. For example, the Truck and Trailer Routing Problem [83], where the capacity of the vehicles is represented using fuzzy constraints, the Fuzzy Maximal Covering Location Problem [32], where the coverage is not considered crisp but fuzzy, and the Perishable Inventory-Routing Problems [43], where the vehicle capacity is also represented in the model as fuzzy constraints.

3.2 Model formulation

A PRP model where the constraint in Eq. 2.3 could be slightly violated can be formulated using fuzzy constraints. The following fuzzy constraint redefines the subset $A' \subseteq A$ containing the arcs that satisfy:

$$A' = \{(i,j) \in A | c_{ij}^k \leq_f M_k \quad \forall k\}. \tag{3.1}$$

The membership function representing the satisfaction degree of the constraint 3.1 is the next piece-wise function:

$$\mu(x) = \begin{cases} 1 & \text{if } x \leqslant M_k, \\ 1 - \frac{x - M_k}{\tau_k} & \text{if } M_k < x \leqslant M_k + \tau_k, \\ 0 & \text{if } x > M_k + \tau_k, \end{cases} \tag{3.2}$$

where $M_k + \tau_k$, $\tau_k \in \mathbb{R}$, is the maximum tolerance allowed for the criterion F_k. With greater tolerance, the problem is more relaxed. In other words, the set of arcs that satisfy the constraint will be larger. That way, it could be possible to find a better solution, a solution with less cost, by relaxing the constraints.

The resulting model of the PRP with fuzzy constraints is formulated as:

$$\min \quad \sum_{(i,j)\in A'} \sum_{k=1}^{|F|} x_{ij} c_{ij}^k w_k,$$

s.t.

$$\sum_j x_{ij} - \sum_j x_{ji} = \begin{cases} 1 & \text{if } i = s, \\ 0 & \text{if } i \neq s,t, \\ -1 & \text{if } i = t, \end{cases}$$

$$A' = \{(i,j) \in A | c_{ij}^k \leq_f M_k \quad \forall k\}, \tag{3.3}$$

$$\sum_{k=1}^{|F|} w_k = 1 \quad \textit{and} \quad 0 \leq w_k \leq 1 \quad \forall k,$$

$$x_{ij} \in \{0,1\} \quad \forall (i,j) \in A'.$$

3.3 Solution approach

Following the parametric approach proposed in [94], the fuzzy problem is transformed into a set of crisp problems. Each crisp problem is obtained using an α-cut with $\alpha \in [0,1]$ that indicates the satisfaction degree of the fuzzy constraint. With this approach the PRP with fuzzy constraints is transformed into a set of PRP_α where the solutions obtained with every value of α constitute the solution of the PRP with fuzzy constraints.

The equation 3.4 describes the α-cuts obtained from the constraint 3.3 of the PRP with fuzzy constraints model, following the membership function 3.2:

$$A'_\alpha = \{(i,j) \in A | c_{ij}^k \leq M_k + \tau_k(1-\alpha) \quad \forall k\}. \tag{3.4}$$

Equation 3.4 limits the arcs available on the graph (the subsets A'_α). It is clear by definition that the arcs in A'_α for an α-cut will be on the A'_α for every α-cut with lower values of α, that is:

$$A'_{\alpha_1} \subseteq A'_{\alpha_2} \iff \alpha_1 \geq \alpha_2.$$

In other words, the subset A' grows (includes more arcs) while the satisfaction degree is reduced.

The problem can be solved by conventional optimisation methods for each value of α. Note that when $\alpha = 1$ the problem is equal to the PRP with crisp constraints, whereas as α takes lower values, the problem is being relaxed in the way that more arcs on the graph satisfy the constraints.

3.3.1 Examples

The examples shown next solve the PRP with fuzzy constraints in a simplified pedestrian scene defined by the set of criteria $F = \{distance, upward\,slope\}$. The fuzzy constraint will limit the maximum upward slope value. Using this simplified scene the solutions will be easier understood and analysed, allowing to better show the behaviour of the PRP with fuzzy constraints.

Each example starts with the definition of the required input values: s, t, W, M, τ and N (the number of α-cuts to be solved). Every PRP_α will be solved with PRoA, the implemented Android application that solves the crisp PRP presented on Sec. 2.4. This application allows to solve the sub-problem generated for each α-cut of the PRP with fuzzy constraints. All examples are located on the city of Granada (Spain).

Example I

This first example solves the PRP with fuzzy constraints in the scene defined by $F = \{distance, upward\,slope\}$, where:

$$s = (37.182438, -3.601203),$$

$$t = (37.180958, -3.594326),$$

$$W = \{1, 0\},$$

$$M = \{\infty, 0.25\},$$

$$\tau = \{-, 0.25\} \text{ and}$$

$$N = 6.$$

The points s and t are located in Granada city centre where there are slopes greater than 0.45. Given those parameters the desired route will has a maximum slope of 0.25 but it will be increased up to 0.50, when the maximum tolerance is allowed.

With the mentioned values, the approach explained will lead to 6 crisp PRP_α (one subproblem per α value) where each set of available arcs is defined according to the α as:

$$A'_\alpha = \{(i,j) \in A | c_{ij}^2 \leq 0.25 + 0.25(1-\alpha)\}.$$

The final solution consists on a set of 6 routes, depicted in Fig. 3.1 with different colour and line thickness. Each route correspond to the solution obtained for the solved α-cut. The user could now choose among them the preferred one.

As it was stated before, when $\alpha = 0$ the problem is more relaxed (the maximum tolerance of the constraint is considered) and it will lead to the route with minimum distance, as expected. The route obtained with $\alpha = 1$ corresponds to the problem with crisp constraints (with no tolerance on the constraints) and shows us the route with greater distance.

In this example, the total distance of the route with crisp constraint ($\alpha = 1$) is 1235.41 m, while the most relaxed problem ($\alpha = 0$) finds a route with total distance of 894.75 m. The distance of the route found at satisfaction degree of 80% ($\alpha = 0.8$) is 1177.11 m, a 4.72% reduction on total distance compared to the route with the least relaxed constraint. Also, with a satisfaction degree of 60% ($\alpha = 0.6$) the distance of the route obtained is reduced to 1008.51 m, a 18.37% reduction on total distance.

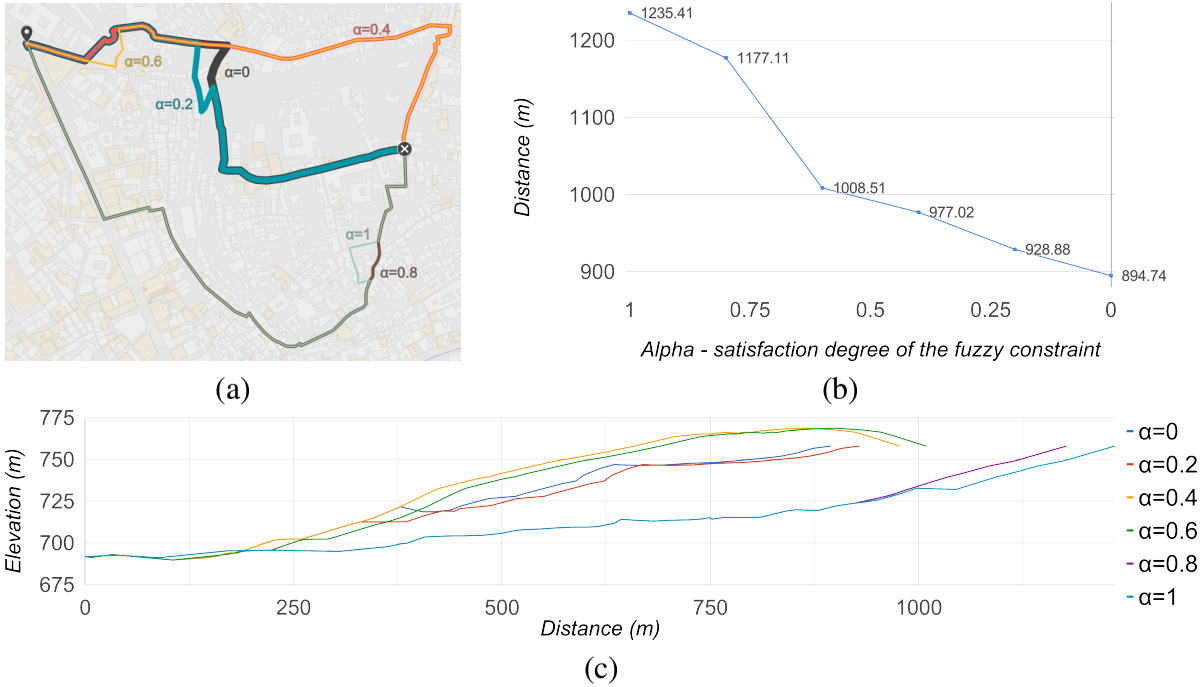


Fig. 3.1 The map in (a) depicts the obtained paths for the example I. The graphic (b) shows the length of the path obtained for every α-cut solved for the example I. In (c), the elevation profile shows the elevation vs distance for every route according to the α value.

Example II

This example solves the PRP with fuzzy constraints in the scene defined by $F = \{distance, upward\,slope\}$ and with parameters:

$s = (37.189529, -3.627628)$,

$t = (37.208121, -3.609622)$,

$W = \{1, 0\}$,

$M = \{\infty, 0.04\}$,

$\tau = \{-, 0.05\}$ and

$N = 6$.

The s and t are located on the outskirts of Granada city where the greatest slope value is lesser than 0.08. The maximum slope on this example is set to 0.04 and it is increased up to 0.09 when the maximum tolerance is considered.

The set of available arcs for each crisp PRP_α is defined as:

$$A'_\alpha = \{(i,j) \in A | c_{ij}^2 \leq 0.04 + 0.05(1-\alpha)\}. \qquad (3.5)$$

Figure 3.2 shows the solution for example II. Again, when $\alpha = 0$ the maximum relaxation of the problem is considered (the maximum tolerance of the constraint) and the route with minimum distance will be obtained. The crisp constraints are considered when $\alpha = 1$ and they will lead to the route with

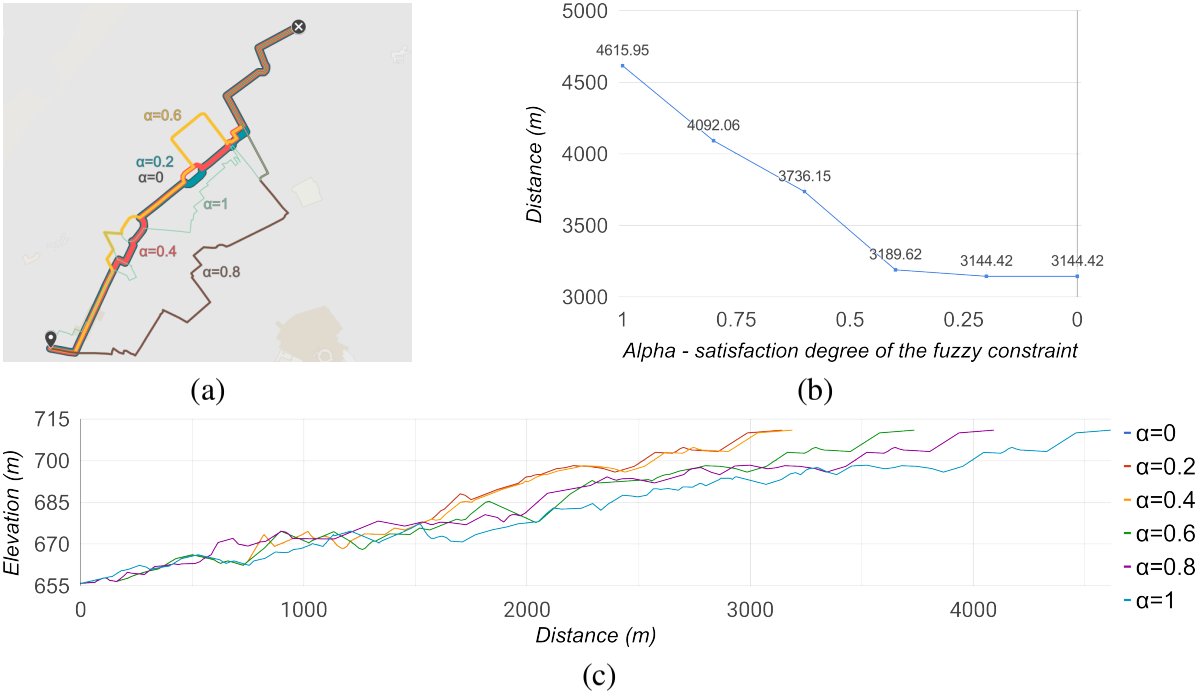


Fig. 3.2 The map in (a) depicts the obtained paths for the example II. The graphic (b) shows the length of the path obtained for every α-cut solved for the example II. The elevation profile in (c) shows the variation on slopes for every route according to the α value.

greater distance. The routes are depicted with different colour and line thickness.

The solution for this example consists on a set of 6 routes, one per each α value. The total distance when $\alpha = 1$ is 4615.95 m while the more relaxed constraint (with $\alpha = 0$) leads to a route of 3144.42 m. With only a little relaxation (with $\alpha = 0.8$) the route's total distance is improved to 4092.06 m, a reduction of 11.35% compared against the route obtained with the least relaxed constraint.

In this example, there are two identical solutions: the routes obtained with $\alpha = 0$ and $\alpha = 0.2$. The duplicity occurs because the sets of available arcs for those different α values are not different enough to find a better route. In other words, there are no arcs with slope value on the interval delimited by their two α-cuts that could improve the final route in terms of distance. This duplicity also implies that every $\alpha \in [0, 0.2]$ would lead into the same solution, which is a non desired situation.

3.4 Adaptive solution approach

The transformation from the fuzzy problem into a set of crisp problems requires the definition of a set of α values, each one leading to a crisp PRP_α problems. In general, the values of α are fixed at the beginning of the process and distributed uniformly in $[0, 1]$, e.g. $\alpha = \{0, 0.25, 0.5, 0.75, 1\}$. However, when dealing

with combinatorial problems, different values of α could lead to the same solution. The solutions' duplicity is also detected in the scene of PRP where it is not possible to guarantee that a different solution is obtained for each α value, see the example shown in Fig. 3.2, where solutions obtained with $\alpha = 0$ and $\alpha = 0.2$ are the same exact route. When solving the PRP, a diverse set of solutions usually has more value, thus solving some α values that will lead to already found solutions is an unnecessary process. In previous example, $\alpha \in [0, 0.2]$ will lead to the same solution.

In that scenario, and when the maximum number of different α-cuts to solve is fixed (due to limited resources), the α values selected could be consciously chosen. Let $N \geq 2$ be the maximum number of different α values to consider or, in other words, the maximum number of PRP_α problems to solve. The proposed method consists on solve PRP_α problems one by one with the idea that the next α to evaluate adapts "on-the-fly" to already known solutions.

The method's pseudocode is described in Alg. 3. First, $\mathscr{H}$, the set of differences in costs from already solved PRP_α, is initialised as an empty set. Then, the first two extreme values of α are solved, $PRP_{\alpha=0}$ and $PRP_{\alpha=1}$. They will lead to solutions with costs $C_{\alpha=0}$ and $C_{\alpha=1}$, respectively. The difference $\Delta C_{0,1} = |C_{\alpha=0} - C_{\alpha=1}|$ is added to $\mathscr{H}$.

As long as the maximum number of PRP_α to solve has not been reached and there is a $\Delta C_{i,j}$ non zero in $\mathscr{H}$, the method proceeds in an iterative way selecting the next $\alpha = m$ the middle

point of the extreme values, l and r, from the greatest $\Delta C_{l,r}$ in $\mathscr{H}$. Then, $PRP_{\alpha=m}$ is solved obtaining $C_{\alpha=m}$. Finally, $\mathscr{H}$ set is updated by adding $\Delta C_{l,m}$ and $\Delta C_{m,r}$ and removing $\Delta C_{l,r}$. However, if there is no non zero element in $\mathscr{H}$, it means that all obtained costs are the same and there will be no α value that can obtain a different solution.

The method initialises by solving $PRP_{\alpha=0}$ and $PRP_{\alpha=1}$. In the first iteration, the middle point is solved obtaining $C_{\alpha=0.5}$ and $\mathscr{H}$ is updated according. Next iteration consists on select α value depending on the differences between $\Delta C_{0,0.5} = |C_{\alpha=0} - C_{\alpha=0.5}|$ and $\Delta C_{0.5,1} = |C_{\alpha=0.5} - C_{\alpha=1}|$. If $\Delta C_{0,0.5} > \Delta C_{0.5,1}$, next value will be $\alpha = 0.25$, while, in other case, $\alpha = 0.75$ will be chosen.

3.4.1 Example

In this example the PRP with fuzzy constraints will be solved with $F = \{distance,\ upward\,slope\}$, following both the non adaptive and the adaptive approaches considering:

$$s = (37.177420, -3.595293),$$

$$t = (37.185850, -3.600150),$$

$$W = \{1, 0\},$$

$$M = \{\infty, 0.04\},$$

$$\tau = \{-, 0.20\} \text{ and}$$

Algorithm 3 Adaptive approach for solving PRP with fuzzy constraints.

1: $N \leftarrow$ maximum number of problems to solve
2: $\mathscr{H} \leftarrow \{\}$
3: Solve PRP_α for $\alpha = \{0,1\}$ obtaining solutions with costs $C_{\alpha=0}$ and $C_{\alpha=1}$
4: $\Delta C_{0,1} \leftarrow |C_{\alpha=0} - C_{\alpha=1}|$
5: $\mathscr{H} \leftarrow \mathscr{H} \cup \{\Delta C_{0,1}\}$
6: $count \leftarrow 2$ ▷ number of already solved PRP_α
7: **while** $count < N$ and $\exists \Delta C_{i,j} \in \mathscr{H}$ with $\Delta C_{i,j} \neq 0$ **do**
8: $\Delta C_{l,r} \leftarrow max\{\mathscr{H}\}$ ▷ the greatest value from active labels
9: $m \leftarrow \frac{l+r}{2}$ ▷ intermediate value
10: Solve $PRP_{\alpha=m}$
11: $count \leftarrow count + 1$
12: $\Delta C_{l,m} \leftarrow |C_{\alpha=l} - C_{\alpha=m}|$
13: $\Delta C_{m,r} \leftarrow |C_{\alpha=m} - C_{\alpha=r}|$
14: $\mathscr{H} \leftarrow \mathscr{H} \cup \{\Delta C_{l,m}, \Delta C_{m,r}\}$
15: $\mathscr{H} \leftarrow \mathscr{H} - \{\Delta C_{l,r}\}$
16: **end while**

$$N = 6.$$

The maximum slope on this example is set to 0.04 and the maximum tolerance on the fuzzy constraint is $M_s + \tau_s = 0.24$. With those values, the set of available arcs for each PRP_α to solve is obtained as:

$$A'_\alpha = \{(i,j) \in A | c_{ij}^2 \leq 0.04 + 0.20(1-\alpha)\}.$$

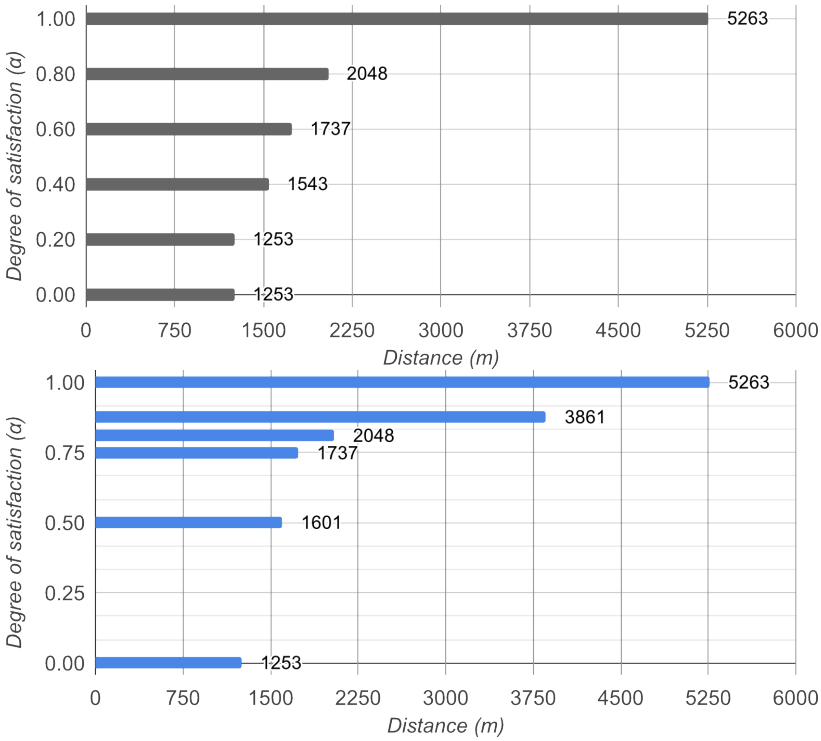


Fig. 3.3 Distances obtained (a) using a non adaptive approach where the solved α-cuts are with $\alpha = \{0, 0.20, 0.40, 0.60, 0.80, 1\}$ and (b) using the adaptive approach where the solved α-cuts are with $\alpha = \{0, 0.5, 0.75, 0.8125, 0.875, 1\}$.

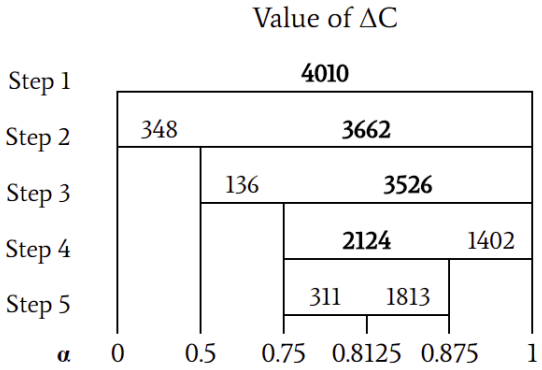


Fig. 3.4 Situation on every step of the adaptive method. On each step the selected α value and active labelled ΔC values are depicted.

Firstly, 6 PRP_α with $\alpha = \{0, 0.2, 0.4, 0.6, 0.8, 1\}$ are solved as typical non adaptive approaches. The solutions obtained for those crisp problems are shown in Fig. 3.3 (a). Secondly, the problem is solved following the adaptive approach. The final distances of each PRP_α problem are shown in Fig. 3.3 (b). One can see that, in both cases, the less restrictive route (with $\alpha = 1$) is longer and with less slopes than the rest. The more tolerance is assumed (with decreasing values of α) the less strict the slopes are and a shorter route can be found.

Figure 3.4 shows the steps on the adaptive approach by illustrating each alpha value chosen and their solutions' cost. At the end of the process, the considered values are $\alpha = \{0, 0.5, 0.75, 0.8125, 0.875, 1\}$. The process stops when the maxi-

mum number of PRP_α problems is reached. However it would be easy to increase that maximum and, if needed, search for more solutions with $\alpha \in (0.8125, 0.875)$ or $\alpha \in (0.875, 1)$, as $\Delta C_{0.8125,0.875} = 1813$ and $\Delta C_{0.875,1} = 1402$ are the greatest values on the distances' difference among the already obtained solutions.

The routes obtained with the non-adaptive approach are shown in Fig. 3.5, while the ones obtained with the adaptive approach are shown in Fig. 3.6. The adaptive approach leads to a solution where every route is different (a perfect scenario) while the non-adaptive approach obtains only five different routes (out of six trials).

It is also worth notice that both methods obtained identical routes under different values of α: the solution from $PRP_{\alpha=0.6}$ (non-adaptive) matched with that from $PRP_{\alpha=0.75}$ (adaptive). Also, $PRP_{\alpha=0.8}$ (non-adaptive) matched with $PRP_{\alpha=0.8125}$ (adaptive), and $PRP_{\alpha=0.2}$ (non-adaptive) has the same solution as $PRP_{\alpha=0}$ (adaptive and non-adaptive).

The adaptive method found two routes (with $\alpha = 0.875$ and $\alpha = 0.5$) not found with the non-adaptive method. The non-adaptive method obtained the route with $\alpha = 0.4$ that is not obtained with the adaptive approach, however, in the map it can be seen that this solution is more similar to other route, the one obtained with $\alpha = 0.6$. The final set of routes is more diverse on the adaptive method. The route obtained with $\alpha = 0.875$ is

key in this aspect because it is an intermediate route between two very different routes (with $\alpha = 1$ and $\alpha = \{0.8, 0.8125\}$).

3.5 Conclusions

A mathematical formulation for the Personalised route problem with Fuzzy Constraints is proposed with the intention to include natural language typical vagueness into the specification of the constraints. Two different approaches are described to solve the PRP with fuzzy constraints, both of them are based on the concept of α-cuts but differ on the way the α values are selected, obtaining a different solution. The first one consists on solving all α-cuts uniformly distributed between 0 and 1 while the second one adds some adaptability defining the next α-cut to solve based on already known solutions from previous α-cuts solved.

The examples solved the PRP with fuzzy constraints in a scene defined by distance and upward slopes criteria posing a fuzzy constraint on the upward slopes. The examples shown the type of information that can be gained from these approaches: for each relaxation level, a potentially new route with a shorter distance can be obtained.

The final solution consists on a set of routes that satisfy the fuzzy constraints with a different degree of satisfaction. However, this is not true in general, as not every PRP with fuzzy constraints will have a set of different routes as a solution, in the same

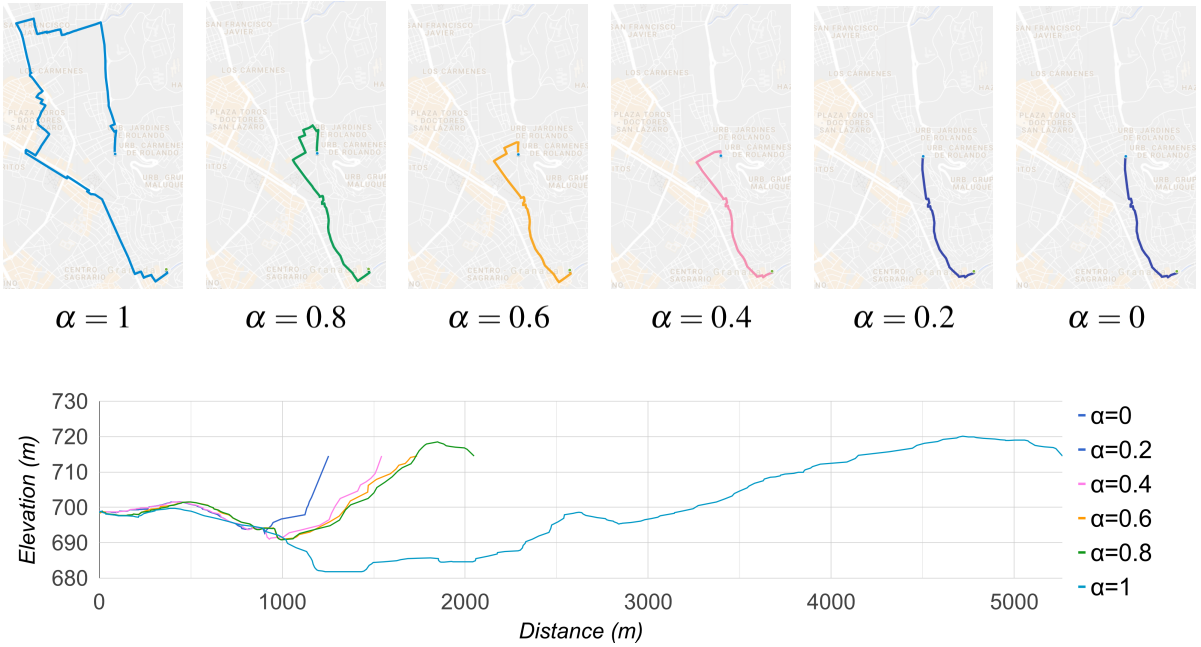


Fig. 3.5 Set of routes obtained with the non-adaptive method and their elevation profile.

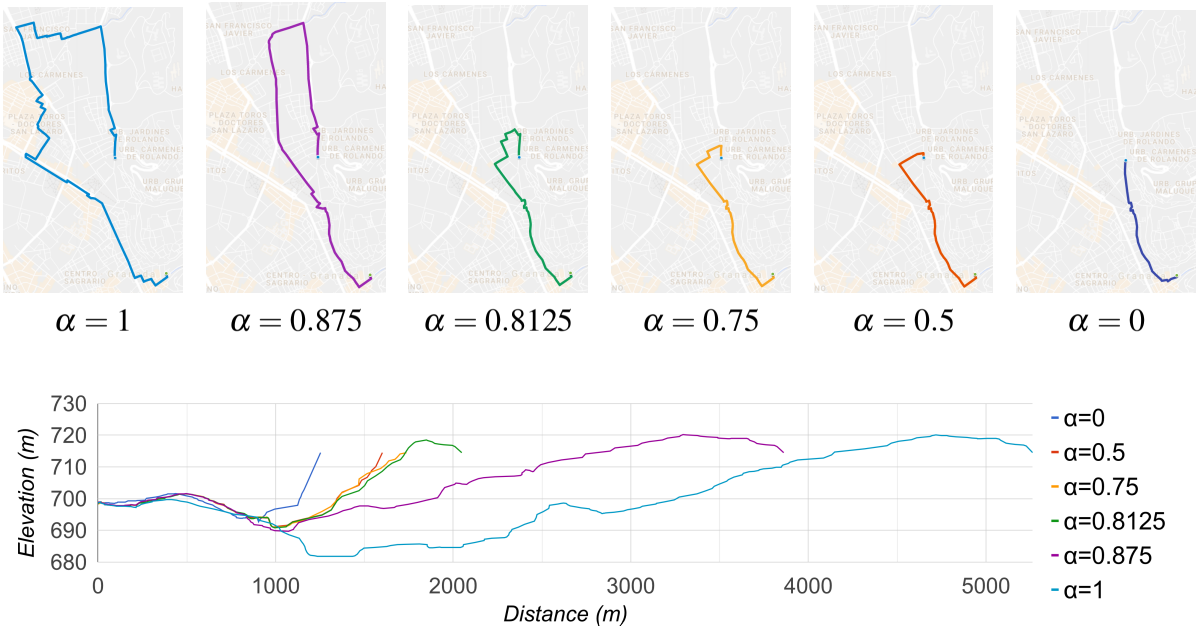


Fig. 3.6 Set of routes obtained with the adaptive method and their elevation profile.

way that a crisp constraint will not modify a solution if the constraint is satisfied by every arc of the solution path. Obviously, the problem could also be not feasible, as happens with crisp constraints. A minimum knowledge of the underlying map is required to properly select the input parameters values that appear on the fuzzy constraint formulation for each α-cut.

The fuzzy constraints are presented and defined based on the pedestrian scene, however, it could be applied on different scenes or problem variations that also deal with constraints. The adaptive approach proposed has been shown to be able to reduce the amount of duplicated solutions and, at the same time, being capable of finding different and more variate solutions solving the same number of PRP_α problems.

Chapter 4

Adaptive maps

The visualisation of specific attributes of the maps like travel time, quality of the road or touristic interest of the path between two points is not achieved with standard maps representations or area cartograms.

Adaptive Maps depict the suitability/difficulty of the paths between points of interest according to each one's needs or preferences taking into account multiple attributes.

In this chapter a model for Adaptive Maps is presented. A generation and a visualisation methods are proposed and five different Adaptive Maps are generated and visualised considering different attributes and preferences.

4.1 Introduction

A map is a depiction that emphasises relationships among elements of some space. The maps are categorised according to the information provided as general cartography or thematic cartography. General cartography maps provide a large amount of general information for widespread public use (e.g. general purpose maps) while thematic cartography maps focus on specific geographic themes and are thought for a specific audience (e.g. population density maps).

The maps can also be categorised as topographic or topological maps depending on if the shape of land surfaces is preserved. Topographic maps represent the shape and features of the surface (e.g. relief maps) while topological maps are diagrams that show simplified information avoiding unnecessary details (e.g. transit maps). In topographic maps the relation between depicted elements is based on the euclidean distance while in topological maps this measure is not the euclidean distance.

Multiple topological maps, also known as schematic maps [2], are constructed to illustrate certain information like in public transit maps [30, 73]. Those maps have the simplification of the network as main objective with the purpose of helping to visualise specific features. Another example of topological map are the area cartograms [76, 28]. In those representations geographic variables are visualised as spatial objects whose size is proportional to certain variable strength.

However, the purpose of "Adaptive Maps" is to help different users to see a "personalised" map reflecting their needs depicting the suitability/difficulty of the paths between *points of interest* (POIs). In contrast to area cartograms, the information in the Adaptive Maps is related to routes (instead of areas) and to the user's preferences instead of certain information. Also, Adaptive Maps consider, simultaneously, multiple attributes like travel time, distance, slopes, etc.

Imagine a pedestrian that has difficulties on walking steeped slopes choosing a route on a map between two points. Then, a relief map and a street map are both required to select the most affordable route. However, a topological map depicting distance and elevation features at the same time could help the pedestrian to directly choose the route based on the distances shown in such supposed topological map.

Now, if a driver is interested on routes without traffic between the same two points, the features have changed and the required topological map will not be the same: it has to be adapted to the new user's preferences. For that purpose, it is required to generate a topological map for each one of the users due to their different preferences.

The "subjective space" [74] is a similar concept. The subjective space was defined as a geographical space such as perceived from a specific subject, like a mental map. In order to generate it, the subject has to previously know the space. More recently, a research on spatiotemporal cognitive map formation [44] is based

on a very similar idea. In contradistinction to those concepts based on mental images, the Adaptive Maps' idea is to show how the space is according to the subject's perception without previously knowing the space itself.

A different way of personalising maps is proposed in [56] where POIs and their connections are highlighted depending on the user's preferences with the objective of showing a map easier to read and understand than a usual standard map. However, the topology of the map is not modified, that is, the underlying map is a standard map, so the users can not have their suitability/difficulty of the paths between POIs depicted on the map, as shown in an Adaptive Map.

In next section a proposal to achieve the Adaptive Map's construction and visualisation is presented.

4.2 Model and generation

An Adaptive Map is a modification of a general purpose map in terms of the user's preferences. An Adaptive Map departs from a general map, a set of POIs, like restaurants, museums or parks, that are of interest for the user, and information of multiple attributes values, e.g. distance among points, slopes among points, quality of the road among points or traffic. The relevance of each attribute is associated with the user preferences. Then, an Adaptive Map represents the suitability/ difficulty (for

a specific user) to travel from each POI to any other. A model formulation for the Adaptive Map is presented below.

The model departs from a directed and finite graph $G = \{N,A\}$, where $N = \{1,...,n\}$ is a finite set of nodes and $A \subseteq N \times N$ is a finite set of directed arcs denoted by (i, j) where $i \neq j$ and $i, j \in N$. The arcs in A represent street segments from a map. Summarising, the proposed model requires:

G a finite graph $G = \{N,A\}$,

i,N indexes and set of nodes of the graph G,

A set of arcs of the graph G denoted as (i, j),

k,F indexes and set of criteria,

ijk,C indexes and set of costs for the arc (i, j) on the criterion $k \in F$, a normalised value that will be minimised,

k,W index and set of weights that represent the importance of minimising each criterion F_k and

m,P index and set of POIs, $P \subseteq N$.

Figure 4.1 shows an example of the adaptive map model where $N = \{n_1,n_2,n_3,n_4,n_5\}$ nodes, $P = \{p_1,p_2,p_3\}$ POIs and $|F| = 2$ criteria.

Both the criteria F and the costs C will be determined by the current scene, which has to be defined before solving the problem. See Sec. 2.3 for a scene definition example.

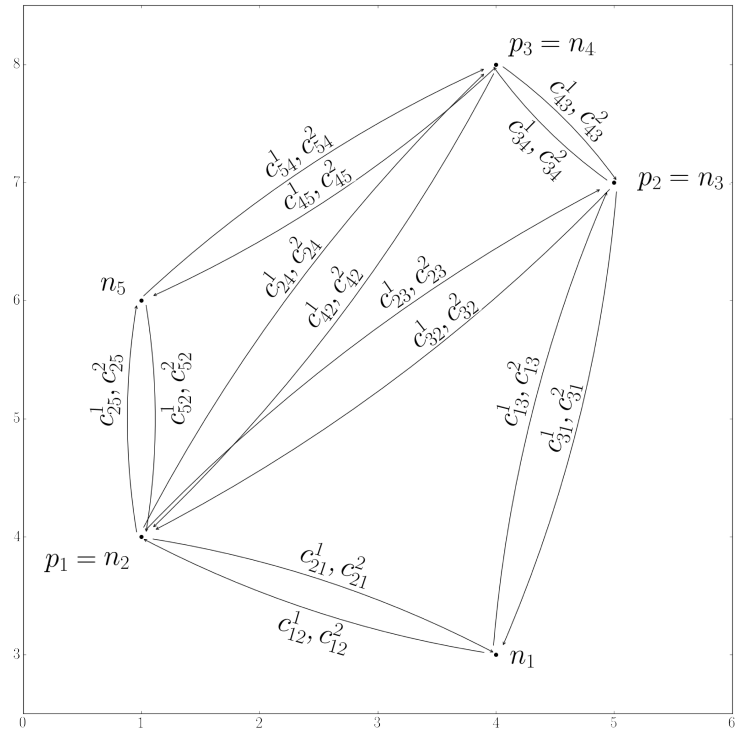


Fig. 4.1 Elements of the model with $|N| = 5$, $|P| = 3$ and $|F| = 2$.

The construction process of the Adaptive Map consists on three steps. First, a *subjective matrix* representing the difficulty for the user to travel between POIs is built. Then, the POIs are relocated in a 2 dimensional space according to that "difficulty". Finally, the map image is modified to fit the POIs' new location. These steps are detailed next.

4.2.1 Generation of a subjective matrix

The subjective matrix is an user dependent matrix that represents the user's suitability/difficulty when travelling between every pair of POIs.

Supposing the user is only interested on the distance criterion, just a single cost associated with the Euclidean distance has to be considered. In this case, and in order to obtain the subjective matrix representing the travel cost for the user, it is necessary to run a shortest path algorithm for every pair of POIs $p_i, p_j \in P$. The travel cost is then interpreted as a measure of suitability/difficulty for the current user.

When a more complex scene with multiple criteria is considered and the user preferences are taken into account, the shortest path algorithm becomes useless. In turn, a PRP is then solved between every pair of POIs in $P \subseteq N$, obtaining the personalised measurement of the path between them. That means that the PRP is solved for every pair of starting and ending points $p_i, p_j \in P$ with $i \neq j$.

The solutions' costs for the solved PRPs are organised as a subjective matrix called M, shown in Table 4.1, where the value $C(p_i, p_j)$ represents some notion of suitability/difficulty that the user may perceive (according to his/her preferences) to traverse the path joining p_i with p_j, where $p_i, p_j \in P$, $i, j = \{1, \ldots, m\}$.

The larger the $C(p_i, p_j)$ value, the greater the effort on following that path for the current user, considering the given prefer-

M	p_1	...	p_i	...	p_m
p_1	0	...	$C(p_1,p_i)$	...	$C(p_1,p_m)$
...	...	...	...	...	...
p_i	$C(p_i,p_1)$	...	0	...	$C(p_i,p_m)$
...	...	...	...	...	...
p_m	$C(p_m,p_1)$	...	$C(p_m,p_i)$	...	0

Table 4.1 Subjective matrix M. The value $C(p_i,p_j)$ stands for the measure of the path joining p_i with p_j.

	x	y
p_1	3	19
p_2	16	20
p_3	13	4
p_4	4	2

Table 4.2 Original location of the POIs.

ences. Note that the subjective matrix M will be different under different sets of preferences.

An example of subjective matrices generation is shown next. The considered initial graph G is depicted in Fig. 4.2 where there are two criteria and $|P| = 4$ POIs located at the positions indicated in Table 4.2.

Supposing that there are 4 users with preferences $W1 = \{W_1 = 1, W_2 = 0\}$, $W2 = \{W_1 = 0, W_2 = 1\}$, $W3 = \{W_1 = 0.5, W_2 = 0.5\}$ and $W4 = \{W_1 = 0.9, W_2 = 0.1\}$ respectively, the obtained graphs illustrating the PRP's costs for each arc and the

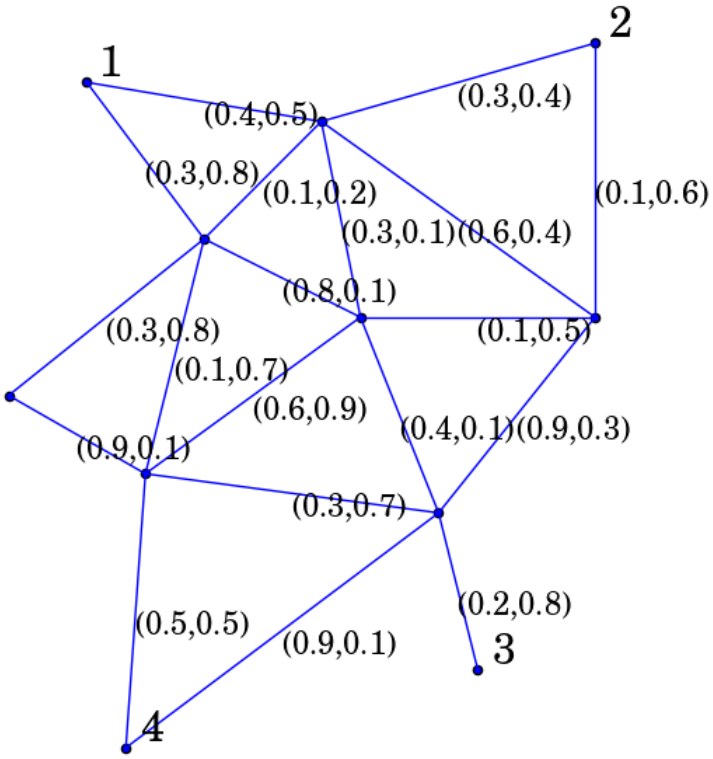


Fig. 4.2 Original graph with two measures. The measurements are normalised.

subjective matrices are shown in Fig. 4.3, Fig. 4.4, Fig. 4.5 and Fig. 4.6 respectively.

Next step will consist on find a new location for the POIs trying to fit the subjective matrix values.

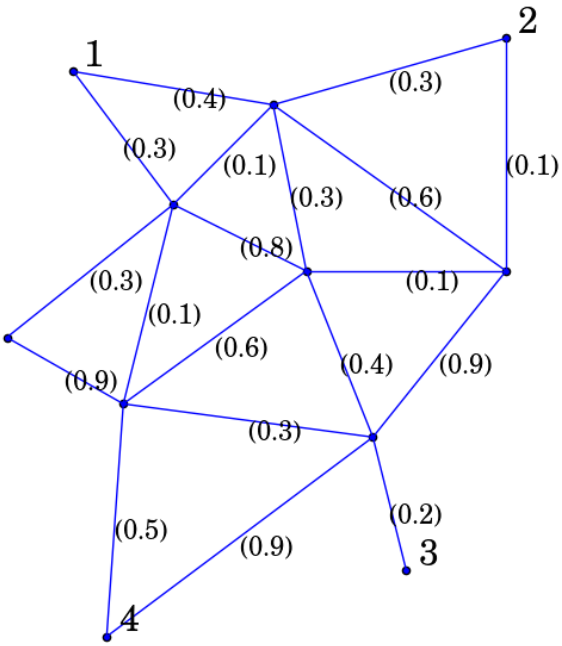
4.2.2 Relocation of POI

As has been previously stated, a map is a depiction that emphasises relationships among elements of some space. At this point, the elements (the POIs) and their relations (in the subjective matrix) are already defined. Now, the visualisation of the Adaptive Map implies two more steps: the relocation of the POIs and the modification of the original map's image. Next, the relocation of the POIs is addressed.

At this stage, the original POIs should be relocated according to the values in the subjective matrix M. This is achieved by finding the set of new locations $Q = \{q_1, q_2, \ldots, q_m\}$, $q_i \in \Re^2$ for the points $p_1, p_2, \ldots, p_m \in P$ in such a way that the distances among q_i, q_j reflects as much as possible $C(p_i, p_j) \in M$.

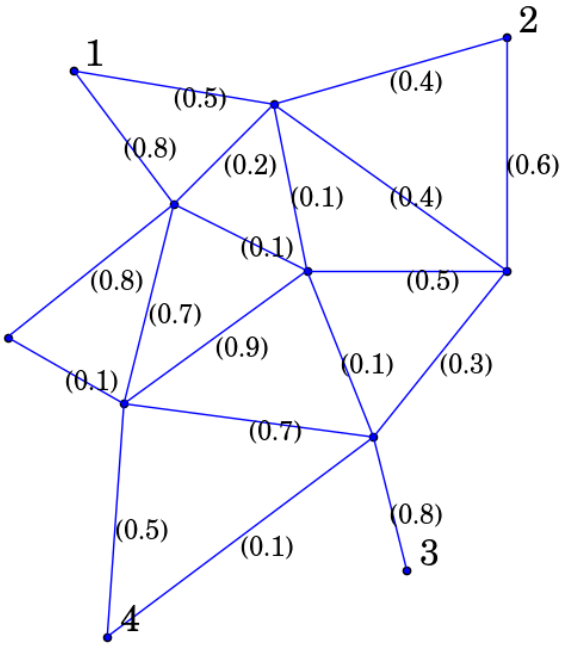
As the path attributes may come from a non-metric space (thus their combination is also a non-metric space), determining Q is a complex problem and, to the best of our knowledge, can not be solved exactly.

In any case, under the hypothesis that an approximated solution may fit the visualisation needs, this problem can be formulated as a Distance Geometry problem [60] as well as a Multidi-



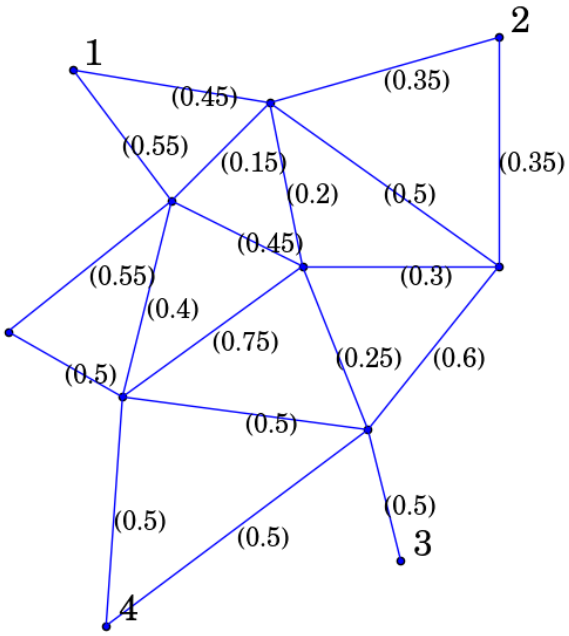
M_1	p_1	p_2	p_3	p_4
p_1	0.0	0.7	0.9	0.9
p_2	0.7	0.0	0.8	1.0
p_3	0.9	0.8	0.0	1.0
p_4	0.9	1.0	1.0	0.0

Fig. 4.3 Graphs illustration for first user with $W1 = \{W_1 = 1, W_2 = 0\}$ the subjective matrix M_1 .



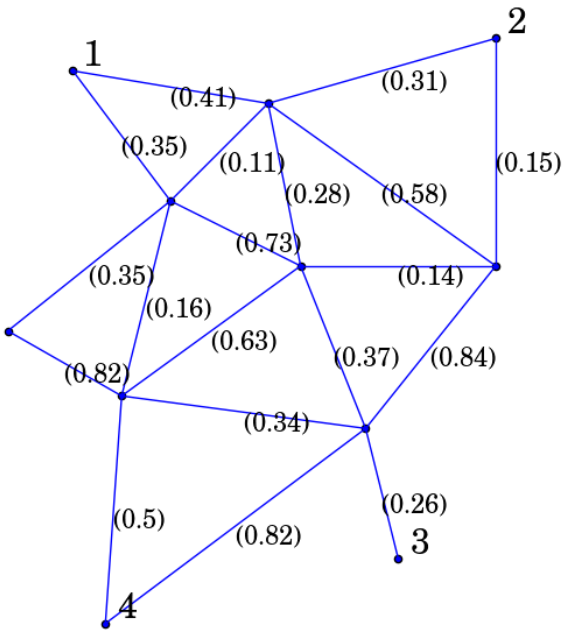
M_2	p_1	p_2	p_3	p_4
p_1	0.0	0.9	1.5	0.8
p_2	0.9	0.0	1.4	0.7
p_3	1.5	1.4	0.0	0.9
p_4	0.8	0.7	0.9	0.0

Fig. 4.4 Graphs illustration for second user with $W2 = \{W_1 = 0, W_2 = 1\}$ and the subjective matrix M_2.



M_3	p_1	p_2	p_3	p_4
p_1	0.0	0.8	1.4	1.4
p_2	0.8	0.0	1.3	1.35
p_3	1.4	1.3	0.0	1.0
p_4	1.4	1.35	1.0	0.0

Fig. 4.5 Graph illustration for third user with $W3 = \{W_1 = 0.5, W_2 = 0.5\}$ and the subjective matrix M_3.



M_4	p_1	p_2	p_3	p_4
p_1	0.0	0.72	1.11	1.01
p_2	0.72	0.0	1.18	1.08
p_3	1.11	1.18	0.0	1.08
p_4	1.01	1.08	1.08	0.0

Fig. 4.6 Graph illustration for forth user with $W4 = \{W_1 = 0.9, W_2 = 0.1\}$ and the subjective matrix M_4.

mensional Scaling (MDS) problem [16]. Due to the availability of software implementations, the solution in this proposal is obtained solving a classical Multidimensional Scaling (classical MDS) problem.

MDS problem departs from a symmetric matrix known as dissimilarity matrix M' (the normalised subjective matrix) and calculates the location of the POI such that the distances between the points are approximately equal to the values on the matrix. The solution from a classical MDS problem is a coordinate matrix that minimises a loss function called *strain* given by:

$$Strain(q_1,\ldots,q_m) = \left(\frac{\sum_{i,j}(b_{i,j} - \langle q_i, q_j\rangle)^2}{\sum_{i,j} b_{i,j}^2}\right)^{1/2} \tag{4.1}$$

where $b_{i,j}$ are coefficients from the matrix $B = -\frac{1}{2}JM'^2J$, $J = I_m - \frac{1}{N}\mathbb{O}$ is a centring matrix and $\mathbb{O}$ a N-by-N matrix of 1.

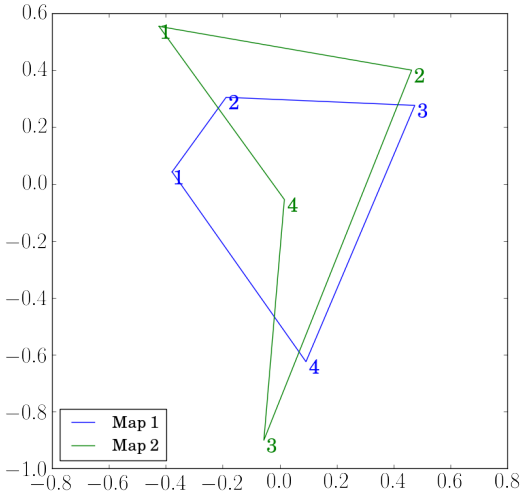
The output of MDS is a set of points Q with the corresponding final locations on the Adaptive Map for the initial POIs, $p_1,\ldots,p_m \in P$. Note that those locations are invariant under reflection, translation and rotation.

As an example, the relocation of the points from Table 4.2 using the subjective matrices shown in Fig. 4.3 and Fig. 4.4 is addressed next. The MDS problem is solved using the *cmdscale*

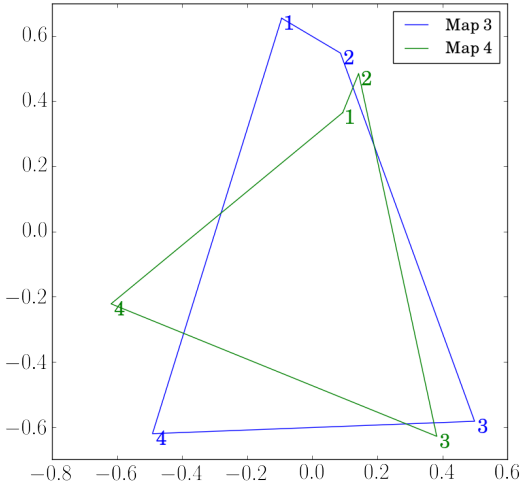
function [72] on *R* [71]. The adapted maps obtained are shown in Fig. 4.7.

If the map is adapted using the subjective matrices M_1 and M_2 (obtained with weights $W1 = \{W_1 = 1, W_2 = 0\}$ and $W2 = \{W_1 = 0, W_2 = 1\}$), the relocation of the POIs is shown in Fig. 4.7 (a). It can be observed that *point 4* and *point 2* are closer for the second user (map 2). This proximity means that the second user has those points "closer" in terms of difficulty, so they can be reached more easily. Also, from *point 1*, first user has *point 2* closer than *point 4* while for the second user both points are at the same distance. It means that the first user will find the path from *point 1* to *point 2* easier than the other one.

The relocation of the points from Table 4.2 using the subjective matrices shown in Fig. 4.5 and Fig. 4.6 is discussed next. From subjective matrices M_3 and M_4 (obtained with weights $W3 = \{W_1 = 0.5, W_2 = 0.5\}$ and $W4 = \{W_1 = 0.9, W_2 = 0.1\}$), the points' relocation results in different solutions, shown in Fig. 4.7 (b). In this case, if the distance from *point 3* to *point 1* is compared to the distance from *point 3* to *point 2*, the first one is longer for third user (map 3) while, for the forth user (map 4), it is shorter. In other words, the third user will consider the path between *point 3* and *point 2* easier, while the forth user finds easier the route between *point 3* and *point 1*.



(a)



(b)

Fig. 4.7 visualisation of the location of the points in the adapted maps (a) *map 1* and *map 2* generated with matrices M_1 and M_2, and (b) *map 3* and *map 4* generated with matrices M_3 and M_4.

4.2.3 Image modification

The final step of the visualisation consists on modifying the original image of the map to relocate the POIs in P at the new locations indicated by $q_i \in Q$, thus obtaining the image of the Adaptive Map.

Firstly, for each POI $p_i \in P$, the coordinates $p_i'[0,1]^2 \in P'$ on the original map image are identified. The coordinates in P' depend on the considered original image.

Secondly, a correspondence is established between the points in P' and Q by mapping each vertex p_i' to q_i and applying a piecewise linear homeomorphism (a bijective continuous function between topological spaces) [31, 58]. At the end, the original image is "distorted" using the q_i locations as anchor points. A way to do this is to obtain a triangulation $T_{p'}$ of P' (a subdivision into triangles) of the image. Then, the image of each triangle from $T_{p'}$ is drawn as a transformed triangle according to the vertices $q_i \in Q$, assuming that the homeomorphism is linear on each triangle. Figure 4.8 illustrates the idea.

It may happen that a triangle superposes a second triangle, in that case, one of the two triangles is faced down (it has an opposite normal vector) and it will not be drawn, as usually done in computer graphics field.

Even though multiple triangulation methods are suitable for this step, the Delaunay triangulation [47, 98] is applied because its properties, explained next.

Fig. 4.8 Modification of an image obtained by mapping the vertices into different positions. In (a), the original image with 4 points in the set P' and a triangulation $T_{p'}$. In (b) and (c) the image's modifications considering two different sets Q of the vertices' final locations.

The Delaunay triangulation $DT_{p'}$ of a set of points P' in 2-dimensional space is a triangulation such that for every circumcircle (the circle which passes through all the vertices of the triangle in $DT_{p'}$) there is no other point in P' inside it.

The Delaunay triangulation has a very important property: it maximises the minimum angle of the triangles. It means that the smallest angle in the Delaunay triangulation is at least as large as the smallest angle in any other triangulation of the points P'. In simpler words, the Delaunay triangulation avoids narrow triangles helping to obtain better results in terms of image quality, which is an important aspect on the Adaptive Maps.

4.3 Adaptive map examples: Granada

In this section different examples are shown to illustrate the usefulness of the proposal. Example A consists on visualising three Adaptive Maps, each one using a single criterion. Example B shows the process of generation and visualisation of Adaptive Maps with multiple criteria.

4.3.1 Example A: one attribute examples

The initial set of POIs consists on 10 points shown in Table 4.3 which are located in the city of Granada (Spain) as illustrated in Fig. 4.9.

Three criteria are taken into account ($|F| = 3$):

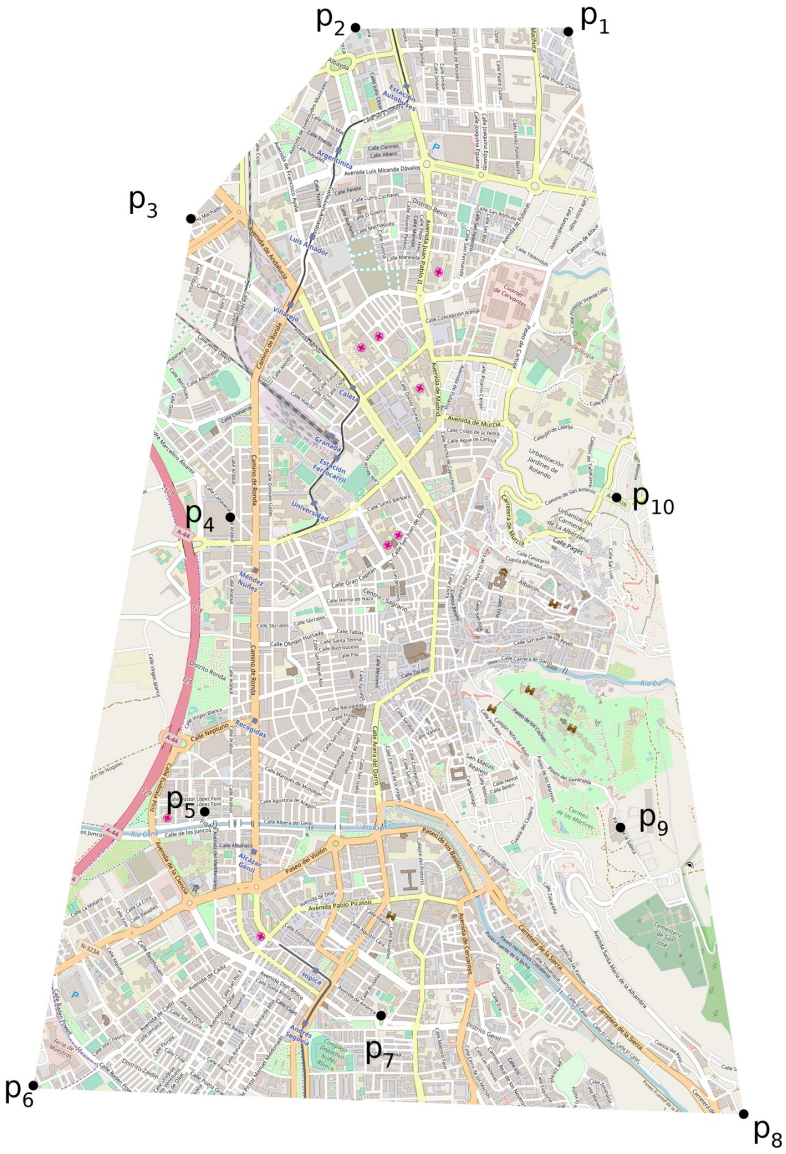


Fig. 4.9 Original map image.

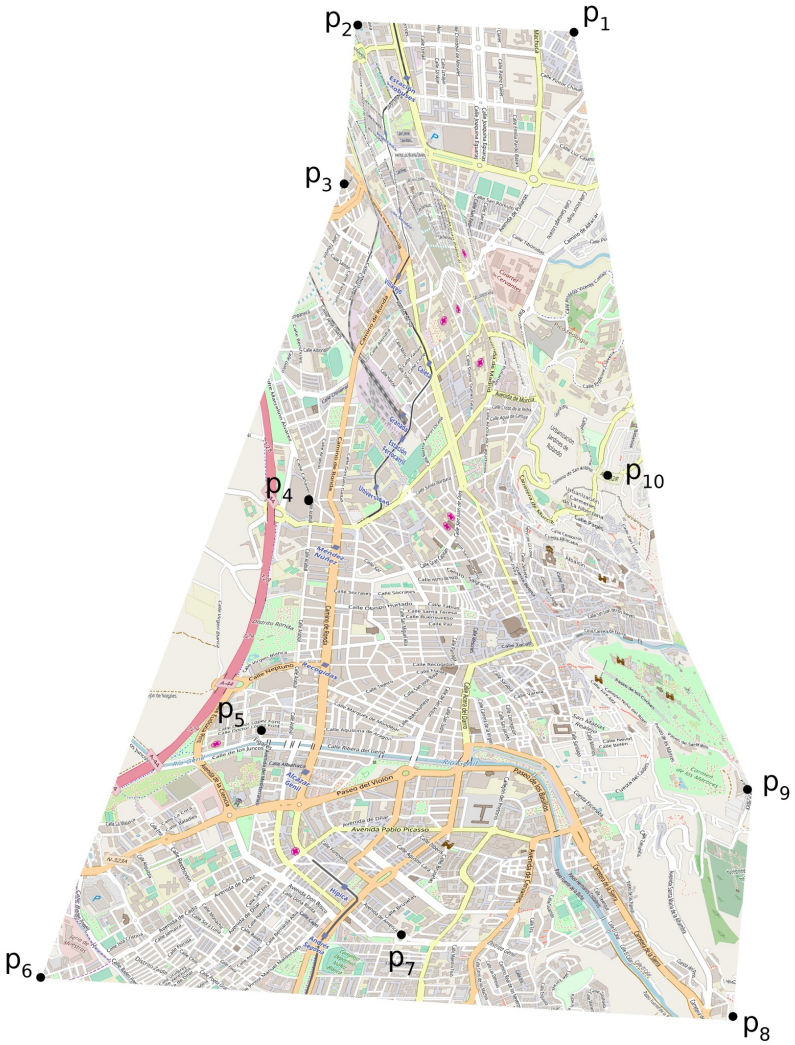


Fig. 4.10 Adaptive Map visualisation for *Adaptive Map 1*, distance on foot.

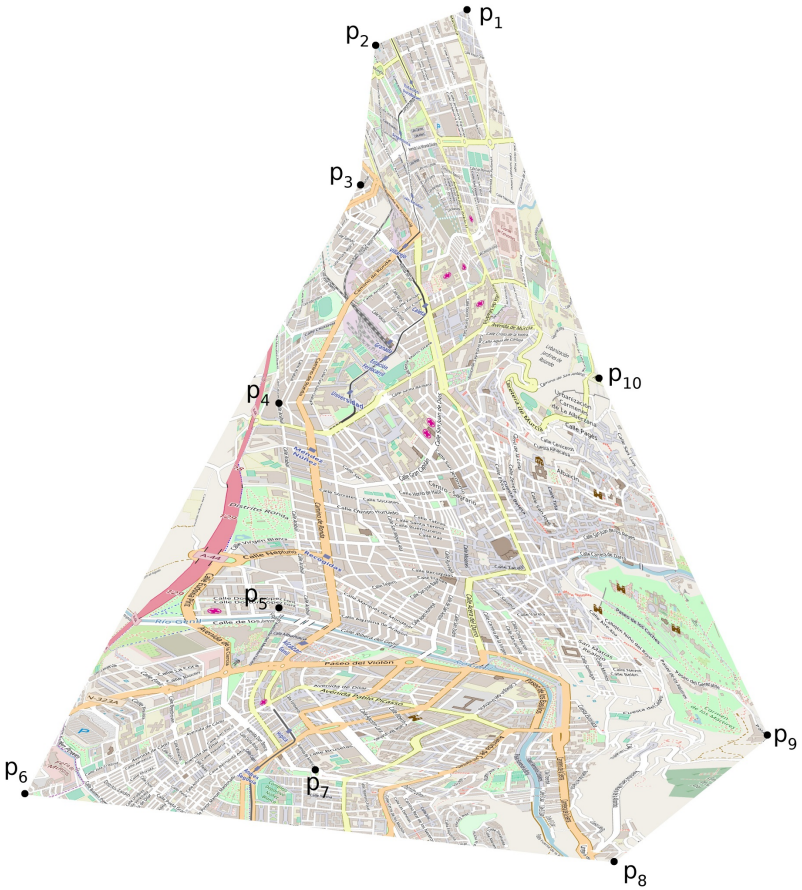


Fig. 4.11 Adaptive Map visualisation for *Adaptive Map 2*, distance on bicycle.

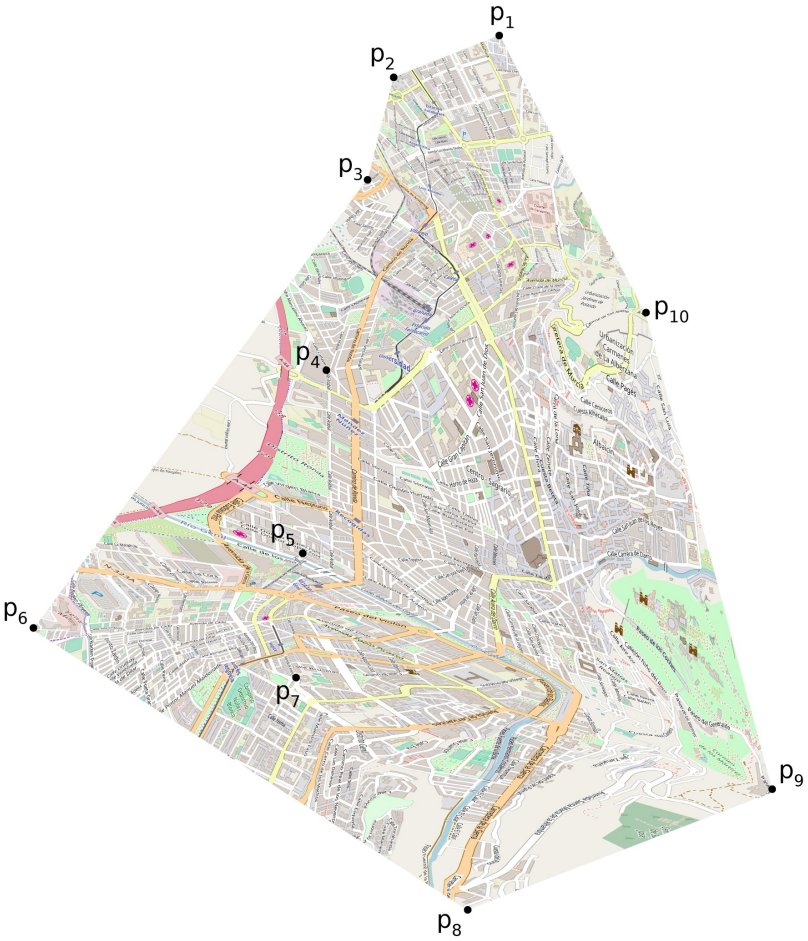


Fig. 4.12 Adaptive Map visualisation for *Adaptive Map 3*, time on bicycle.

	Latitude	*Longitude*
p_1	37.206687	−3.606632
p_2	37.202209	−3.618311
p_3	37.190825	−3.622208
p_4	37.187247	−3.591872
p_5	37.178639	−3.612509
p_6	37.173052	−3.583150
p_7	37.163225	−3.569172
p_8	37.159979	−3.591418
p_9	37.149748	−3.608442
p_{10}	37.165262	−3.606178

Table 4.3 Example A: geographical location of the POI.

1. Distance on foot. The true walking distance from one point to other point.

2. Distance on bicycle. The true distance when cycling from one point to other point. Due to traffic restrictions on roads for bicycles, this distance is usually longer than the distance on foot.

3. Time on bicycle. The time that takes when cycling from one point to other point.

To generate 3 different Adaptive Maps, 3 different weights distributions are set:

- $W_1 = \{1,0,0\}$,

- $W_2 = \{0,1,0\}$ and
- $W_3 = \{0,0,1\}$.

For this example with one single attribute, the subjective matrix M is calculated using the GrapHopper's Matrix API [29]. This API gives the measurements according to one attribute between two geographical points. The subjective matrices M_k and the Q locations are shown in Tables 4.13, 4.14 and 4.15.

The visualisation of every *Adaptive Map k* is achieved as explained in Section 4.2.2. First, the classical MDS problem is solved using the *cmdscale* function on *R* [71] to determine the points' location $q_i \in Q$ on the *Adaptive Map k*. Secondly, the Delaunay triangulation $DT_{p'}$ is calculated using the *Delaunay* function on *Python* [40] and each triangle is drawn by mapping the vertices p'_i into q_i.

The final Adaptive Maps are shown in Fig. 4.9 and Fig. 4.10. POIs' final positions vary from one Adaptive Map to other.

Figure 4.10 shows the final Adaptive Map when considering walking distance. It can be observed that p_1, p_3 and p_5, p_9 look closer than in the original map.

Figure 4.11 shows the adaptive map in terms of distance on bicycle while Fig. 4.12 shows the time on bicycle. As a main difference, the points p_7, p_9 look close in the distance map but far apart in the time map. Also, the point p_{10} is farther away from all points when time on bicycle is considered. Actually, the points p_9 and p_{10} have more elevation than any other POI

M_1	p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8	p_9	p_{10}
p_1	0	1553	2783	3614	4240	5674	6972	6064	7408	5387
p_2	1553	0	1674	3867	2975	5936	7234	6326	6807	4693
p_3	2783	1674	0	3427	2118	5098	6396	5409	5950	3837
p_4	3614	3867	3427	0	2471	2571	4609	3745	5222	3292
p_5	4240	2975	2118	2471	0	3600	4684	3172	3840	1727
p_6	5674	5936	5098	2571	3600	0	2569	3076	5083	3416
p_7	6972	7234	6396	4609	4684	2569	0	2692	4966	3903
p_8	6064	6326	5409	3745	3172	3076	2692	0	2415	1698
p_9	7408	6807	5950	5222	3840	5083	4966	2415	0	2303
p_{10}	5387	4693	3837	3292	1727	3416	3903	1698	2303	0

(a)

	x	y
q_1	0.124	0.488
q_2	−0.075	0.498
q_3	−0.086	0.351
q_4	−0.117	0.060
q_5	−0.161	−0.150
q_6	−0.362	−0.375
q_7	−0.033	−0.336
q_8	0.269	−0.414
q_9	0.286	−0.204
q_{10}	0.155	0.081

(b)

Fig. 4.13 Example A: (a) subjective matrix M_1, (b) locations of points Q.

M_2	p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8	p_9	p_{10}
p_1	0	1791	3561	4106	4228	6889	7403	7260	8082	5822
p_2	2136	0	2212	5009	3498	7750	8264	6915	8075	5423
p_3	2915	2240	0	4637	2190	6474	6988	5562	6605	4308
p_4	3968	4346	3496	0	2915	4978	5492	5349	6172	3912
p_5	4074	3223	2313	3239	0	4780	5294	3572	4394	2080
p_6	6833	6546	5747	2681	4731	0	2543	3428	5198	4188
p_7	7614	7327	6529	5555	5362	2801	0	3347	5117	4527
p_8	6491	6282	5349	4432	3207	4319	2980	0	2393	1849
p_9	8044	7193	6227	6317	4118	6922	5569	2732	0	2458
p_{10}	5655	4803	3870	4055	1728	4510	4457	2079	2672	0

(a)

	x	y
q_1	0.064	0.506
q_2	−0.034	0.468
q_3	−0.051	0.320
q_4	−0.139	0.084
q_5	−0.138	−0.135
q_6	−0.409	−0.335
q_7	−0.097	−0.308
q_8	0.222	−0.408
q_9	0.387	−0.274
q_{10}	0.207	0.109

(b)

Fig. 4.14 Example A: (a) subjective matrix M_2, (b) locations of points Q.

M_3	p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8	p_9	p_{10}
p_1	0	392	738	1423	899	2137	1949	1694	1838	1320
p_2	624	0	471	1652	743	2327	2139	1538	1710	1151
p_3	831	526	0	1637	503	2127	1939	1325	1420	938
p_4	967	969	786	0	640	1656	1468	1213	1357	839
p_5	1086	765	559	1320	0	1675	1488	856	999	468
p_6	1732	1605	1402	1434	1091	0	605	826	1202	949
p_7	2081	1953	1750	2023	1321	1290	0	748	1124	997
p_8	1621	1474	1277	1563	750	1585	875	0	490	401
p_9	1924	1603	1351	1986	880	2113	1413	598	0	535
p_{10}	1491	1170	973	1538	447	1636	1195	514	590	0

(a)

	x	y
q_1	0.093	0.431
q_2	−0.016	0.388
q_3	−0.042	0.284
q_4	−0.084	0.087
q_5	−0.108	−0.101
q_6	−0.385	−0.178
q_7	−0.117	−0.228
q_8	0.060	−0.465
q_9	0.373	−0.345
q_{10}	0.243	0.146

(b)

Fig. 4.15 Example A: (a) subjective matrix M_3, (b) locations of points Q.

	Latitude	*Longitude*
p_1	37.169053	−3.597
p_2	37.167762	−3.596
p_3	37.167129	−3.590
p_4	37.168013	−3.589
p_5	37.169709	−3.593

Table 4.4 Example B: geographical location of the POIs.

and the paths leading to those points consist on zigzag roads. That makes points p_9 and p_{10} harder to reach under mostly all considered attributes.

4.3.2 Example B: multiple attributes examples

In this example two Adaptive Maps are generated and visualised departing from a map located on Granada (Spain) with 5 POI. Point locations are shown in Table 4.4. In this case, the subjective matrices M_1 and M_2 are calculated as explained in Section 4.2.1: a PRP associated to each $C(p_i, p_j)$ value is solved using the software PRoA (an acronym for Personalised Route Assistant), an Android application available on Google Play [84].

Four criteria are considered, $|F| = 4$:

1. Distance on foot. The real distance when walking from one point to other point.

2. Upward slopes. The upward inclination of the street.

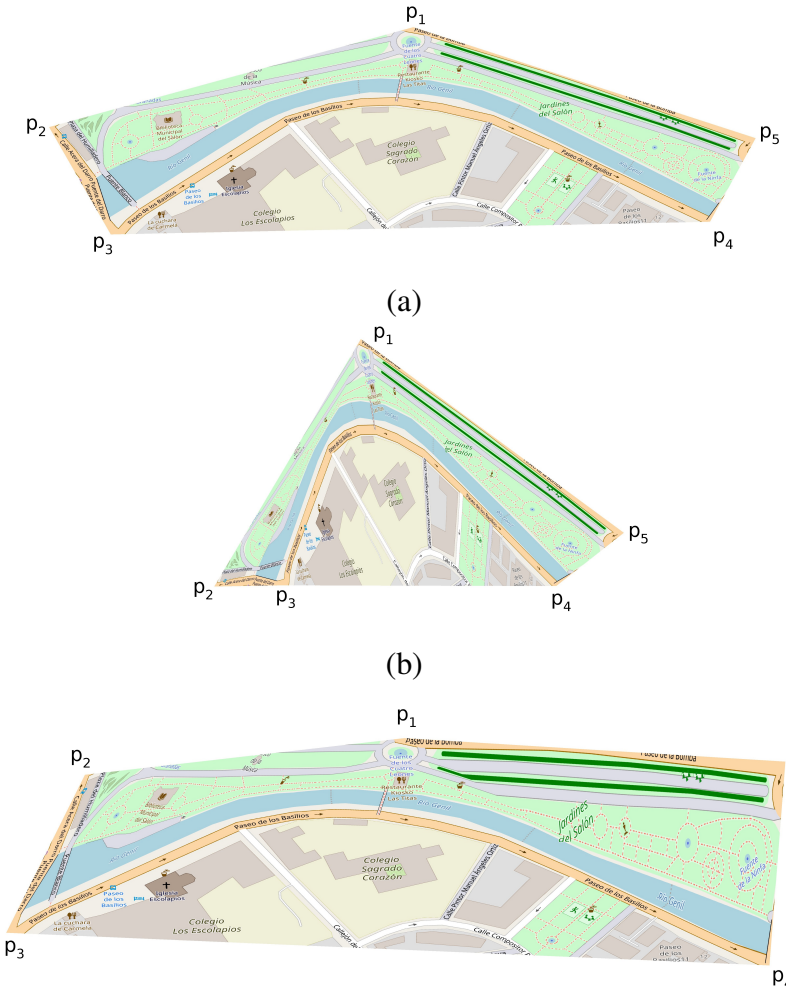


Fig. 4.16 Adaptive Maps visualisation. In (a) the general purpose map, (b) *Adaptive Map 1*, reducing upward, downward slopes and distance, and (c) *Adaptive Map 2*, reducing motor vehicles and distance.

M_1	p_1	p_2	p_3	p_4	p_5
p_1	0.000	0.111	0.486	0.581	0.411
p_2	0.108	0.000	0.392	0.462	0.359
p_3	0.480	0.390	0.000	0.071	0.428
p_4	0.604	0.514	0.125	0.000	0.512
p_5	0.406	0.371	0.442	0.435	0.000

(a)

	x	y
q_1	-0.271	-0.087
q_2	-0.174	-0.085
q_3	0.217	-0.086
q_4	0.314	-0.010
q_5	-0.062	0.266

(b)

Table 4.5 Example B: (a) subjective matrix M_1, (b) locations of points Q.

3. Downward slopes. The downward inclination of the street.

4. Motor vehicles areas. This attribute means that the user wants to reduce areas with motor vehicles, preferring pedestrian areas like paths or parks.

Two different Adaptive Maps are generated by setting 2 different weights distributions:

- $W_1 = \{0.1, 0.45, 0.45, 0\}$ and
- $W_2 = \{0.3, 0, 0, 0.7\}$.

The weights distribution W_1 for the *Adaptive Map 1* has the following meaning: it is equally important to minimise attributes 2 and 3 (the user does not want slopes) and it is somehow important to reduce the distance. Passing through motor vehicles areas is irrelevant.

M_2	p_1	p_2	p_3	p_4	p_5
p_1	0.000	0.416	1.004	1.022	0.529
p_2	0.405	0.000	1.125	1.129	0.634
p_3	1.013	1.104	0.000	0.374	0.636
p_4	1.037	1.141	0.389	0.000	0.661
p_5	0.533	0.636	0.624	0.553	0.000

(a)

	x	y
q_1	−0.457	0.083
q_2	−0.568	−0.140
q_3	0.512	−0.186
q_4	0.534	0.102
q_5	−0.005	0.135

(b)

Table 4.6 Example B: (a) subjective matrix M_2, (b) locations of points Q.

The weights distribution W_2 for the *Adaptive Map 2* says that the user does not care about slopes, but it is very important to reduce the motor vehicles zones and the walking distance should be also taken into account.

For each weight distribution W_k, the subjective matrix M_k is calculated by solving a PRP for every pair of points $p_i, p_j \in P$, with $i \neq j$ (see Section 4.2.1). The subjective matrices M_k and the Q locations, for *Adaptive Map 1* and *Adaptive Map 2*, are shown in tables 4.5 and 4.6, respectively.

Once the matrix M_k is calculated, the visualisation of the *Adaptive Map k* is achieved as explained in Section 4.2.2. In this example the Delaunay triangulation $DT_{p'}$ consists on 3 triangles.

The final Adaptive Maps are shown in Fig. 4.16. On top, the original map appears. In (b) the *Adaptive Map 1* (considering upward, downward slopes and distance) brings points p_3 and p_4 closer, because they have similar elevation and the path between them barely has slopes. In the *Adaptive Map 2* (obtained avoiding

areas with motor vehicles and minimising distance), shown in (c), the points p_2 and p_5 are closer because the path between them consists on pedestrian zones.

4.4 Conclusions

In this chapter, the concept of Adaptive Maps has been presented and its model has been described. Adaptive Maps are useful for representing specific attributes like *walking* or *cycling time* that standard maps based on Euclidean distance are not able to represent.

From an operational point of view, a method to generate and visualise these Adaptive Maps were also proposed. Examples using single and multiple attributes showed the benefits of the proposal.

Chapter 5

Solutions of interest identification

The result of a multiobjective or a many objective optimisation problem is a large set of non-dominated solutions. Once the Pareto Front (or a good approximation of it) has been found, then providing the decision maker with a smaller set of "interesting solutions" is a key step.

Here the focus is on how to select such a set of Solutions of Interest which, in contrast to previous approaches that relied on geometrical features, is carried out considering the Decision Maker's preferences. The proposed *a posteriori* approach consists in representing the solutions as intervals that depend on the decision maker's preferences. The solutions are then compared

and filtered according to their corresponding intervals, using a recently proposed possibility degree formula.

5.1 Introduction

Most of the problems arising in the current social and technological context require the simultaneous optimisation of several conflicting objectives. These multiobjective (and many objective) optimisation problems (MOPs, in what follows) are approximately solved through specific optimisation algorithms that try to obtain a set of solutions which hopefully are close to the so-called true Pareto Optimal set.

Research on these algorithms (mainly those belonging to the class of evolutionary algorithms) has been prolific in the last decades. Despite their type, all of them rely on the fact that "*the ultimate goal of multiobjective optimisation is to help the decision maker find solutions that meet, at most, his/her preferences*" [51]. In this context, the motivation for this work comes from the following observation: *while the Pareto Front of a given MOP is unique, the selected solution will depend on the preferences of the Decision Maker (DM).*

The relation between the DM's preferences and MOPs has many facets that can, essentially, be grouped into two sets. First, the preferences can be incorporated in the solving algorithm (e.g. using goals, weights and reference vectors) thus guiding the

searching algorithm to specific regions of the solutions' space. These methods can be classified as *a priori* or *progressive* approaches [15].

Secondly, the preferences can be used after the optimisation process to help the DM selecting a solution. These approaches are known as *a posteriori* and consist, for example, in applying some techniques to help the DM to select his/her most preferred solutions [11]. In this case, the DM does not need to provide any preference information prior to the search.

A crucial point here is the number of solutions that the DM will have to manage. As stated in [15]:

> "The number of elements of the Pareto optimal set that tends to be generated is normally too large to allow an effective analysis from the DM."

Moreover, in [18], one can read:

> "The DM is interested in discovering only the zone of the Pareto front corresponding to his/her preferences, instead of the whole Pareto front. It is essential to provide the DM with a small number of satisfactory alternatives, due to the human cognitive limitations [...]."

And a similar conclusion is given in [55]:

> "The empirical investigation revealed dysfunctional effects of information overload if the respondents were provided with ten or more alternatives in the choice set [...]."

Therefore, there is a need to develop approaches to help the DM to find this set of *Solutions of Interest* (SOIs, in what follows) that can be vaguely defined as those solutions that are preferred by the DM [6]. Several measures were proposed to identify the SOIs, but they mainly focus on a more geometric interpretation of the solutions in the Pareto Front, losing the DM's perspective.

The aim of this chapter is to present an approach to identify a set of SOIs from a large set of (already known) non-dominated solutions taking the DM's preferences into account, which is a clear distinction from existing methods. Thus, in the context of multiobjective optimisation, this is an *a posteriori* approach. The DM's preferences are given as a linear ordering of the objectives, therefore partially avoiding the issue recognised in [96]: "*in many cases the user does not have a clear preference when little knowledge about the problem is available*".

While other approaches search for SOIs on an n-dimensional objectives space, this proposal relies on comparing solutions in a bi-dimensional space using a possibility distribution function whose values are later used to filter out and reduce the number of SOIs.

5.1.1 Multiobjective optimisation

Basic definitions associated with MOPs are recalled below. Definitions are adapted from [15], considering maximisation problems.

Definition 1 (Multiobjective optimisation Problem, MOP): A MOP is defined as maximising $F(x) = (f_1(x), \ldots, f_n(x))$ subject to $g_i(x) \leq 0$, $i = 1, \ldots, p$, and $h_j(x) = 0$, $j = 1, \ldots, q$, $x \in \Omega$. A MOP solution maximises the components of a vector $F(x)$ where x is a k-dimensional decision variable vector $x = (x_1, \ldots, x_k)$ from some universe Ω. It is noted that $g_i(x) \leq 0$ and $h_j(x) = 0$ represent constraints that must be fulfilled while maximising $F(x)$. Ω contains all possible x that can be used to satisfy an evaluation of $F(x)$.

The term many objective optimisation problem is applied when the number of objectives is greater than three.

Definition 2 (Pareto Dominance): A vector $u = (u_1, \ldots, u_n)$ is said to dominate another vector $v = (v_1, \ldots, v_n)$ (denoted by $u \succeq v$) if, and only if, u is partially greater than v, i.e., $\forall i \in 1, \ldots, n, u_i \geq v_i \wedge \exists i \in 1, \ldots, n : u_i > v_i$.

Definition 3 (Pareto Optimal Set): For a given MOP, $F(x)$, the Pareto Optimal Set, $\mathscr{P}^*$, is defined as:

$$\mathscr{P}^* := \{x \in \Omega \,|\, \neg\exists x' \in \Omega \quad F(x') \succeq F(x)\}.$$

Definition 4 (Pareto Front): For a given MOP, $F(x)$, and Pareto Optimal Set, $\mathscr{P}^*$, the Pareto Front $\mathscr{PF}^*$ is defined as:

$$\mathscr{PF}^* := \{u = F(x) | x \in \mathscr{P}^*\}.$$

5.1.2 On detecting solutions of interest

There are two pathways to provide the DM with a reduced set of non-dominated solutions. The first one involves incorporating the DM's preferences in the solving algorithm thus guiding the searching algorithm to a "region of interest" (ROI); and secondly, to use some strategy *after* the optimisation process to help the DM in selecting a solution.

Methods within the first set require the DM to describe his/her preferences as desired goals, reference points or a pre-ordering on the objectives prior to the search. These *a priori* approaches have been widely used in the past. The interested reader can check some recent reviews on the topic [96, 53, 51]. However, as has been recently recognised in [51], the definition of ROI is vague: it could be any region of the Pareto Front, controlling its size is far from trivial, and just concentrating on a pre-defined ROI

can lead the user to lose some relevant information regarding the boundaries of the Pareto Front. To overcome these problems, the authors in [51] propose a systematic way to incorporate the DM's preference information, either a priori or interactively, based on a nonuniform mapping scheme of the reference points that guide the search. The DM is required to define an expected value on each objective.

Secondly, the reduction of a Pareto Front (obtained by any means) to a smaller set of "diverse solutions" can be attained using some sort of geometric interpretation. Several approaches exist for detecting SOIs in multiobjective problems with more than two objectives [7].

The first such approach is the maximum convex bulge [20]. It is a purely geometric approach that selects a subset of those located in the "maximum bulge" of the Pareto Optimal Front as SOIs. The second one is the hypervolume contribution [103]. The SOIs are those that produce the maximum gain in the hypervolume. It is computationally expensive and it is also a purely geometric approach. The third one is the local curvature [6] approach. Some curvature is calculated from a given solution and its neighbourhoods, and those with the highest curvature are considered more preferable. The whole approach may fail if the Pareto Front is neither continuous nor symmetric. The expected marginal utility (EMU) measure [12], the fourth approach, is aimed at identifying the so-called "knee solutions". The knee is defined as a solution on the tradeoff surface, where significant

compromise needs to be made in at least one objective in order to obtain small gains in another. This is a stricter definition than "solution of interest". Although EMU seems to work well with 2-3 objectives, it is observed that "*as the number of objectives increases, the proportion of solutions with nonzero EMUs decreases and there are very few solutions with unique EMU values*" [7]. To overcome this limitation, the authors in [7] propose using recursive calculations of EMU obtaining K unique sparse SOIs.

Some aspects are common in these works: first, as no information regarding the DM's preferences is required, their authors proposed some metrics to detect solutions that are "interesting" from a geometrical point of view; and second, as these metrics focus on different features, they may end up suggesting a completely different set of solutions.

5.1.3 Interval numbers comparison

The proposed SOIs identification method consists on assigning an interval to every solution in the Pareto Front, thus solutions will be compared in terms of their corresponding intervals. Comparing and ranking interval numbers is a widely studied topic. Although reviewing the methods in the literature is out of the scope of this proposal, some references are given in order to offer a general view of the topic.

Since the initial work from Moore [59] on interval arithmetic, many approaches have been proposed, ranging from establishing

order relations between interval numbers (which are summarised in [41]) to ranking methods that depend on probabilistic or fuzzy concepts [49, 77].

In the present work, the focus is on using possibility distributions for interval numbers comparisons, following the approach presented in Liu et al. [52]. The authors propose a formulation of a possibility degree to compare (and thus rank) intervals depending on a function that reflects the attitude, idea and knowledge of the DM. Moreover, under certain conditions, Liu's proposal is able to capture the behaviour of other approaches.

Let $A = [a_l, a_r]$, $B = [b_l, b_r]$ be two nonnegative interval numbers with $a_l, a_r, b_l, b_r \in R_0^+$. The possibility degree of A being greater than B, namely $P(A \geq B)$, proposed in [52] is defined as follows:

1. if $A \cap B = \emptyset$,

$$P(A \geq B) = \begin{cases} 0 & a_r \leq b_l \\ 1 & a_l \geq b_r \end{cases} \tag{5.1}$$

2. if $A \cap B \neq \emptyset$,

$$P(A \geq B) = \frac{\int_{b_l}^{a_r} f(x)dx}{\int_{b_l}^{a_r} f(x)dx + \int_{a_l}^{b_r} f(x)dx}, \tag{5.2}$$

where $f(x)$ describes the attitude of the DM.

Three typical functions for modelling the DM's attitudes are suggested: $f(x) = c$ a neutral attitude, $f(x) = 1/x$ a pessimistic attitude[1] and $f(x) = \sqrt{x}$ an optimistic attitude.

When $A \cap B \neq \emptyset$ the possibility degree for each attitude is calculated as follows:

$$P(A \geq B) = \frac{a_r - b_l}{a_r - a_l + b_r - b_l} \tag{5.3}$$

$$P(A \geq B) = \frac{\ln a_r - \ln b_l}{\ln a_r - \ln a_l + \ln b_r - \ln b_l} \tag{5.4}$$

$$P(A \geq B) = \frac{a_r\sqrt{a_r} - b_l\sqrt{b_l}}{a_r\sqrt{a_r} - a_l\sqrt{a_l} + b_r\sqrt{b_r} - b_l\sqrt{b_l}} \tag{5.5}$$

Equation 5.3 for neutral attitude, Eq. 5.4 for pessimistic attitude and Eq. 5.5 for optimistic attitude.

The previous formulation with a neutral attitude (Eq. (5.3)) is able to capture the behaviour of other approaches [19, 26, 80, 97, 101]. Also, the formulations proposed in [102, 106] are equivalent to some of the above [50]. That means that a proper use of the formulation proposed by Liu et al. allows the same results as other methods to be obtained while, at the same time, the interval comparison is easily modified to consider different DM's attitudes.

[1] For certain attitude functions like $f(x) = 1/x$ the intervals must be defined in R^+.

	$P(A \geq B = [0.3, 0.7])$		
	Neutral	Optim.	Pessim.
$A_0 = [0.1, 0.2]$	0	0	0
$A_1 = [0.1, 0.6]$	0.33	0.35	0.26
$A_2 = [0.3, 0.6]$	0.43	0.42	0.45
$A_3 = [0.3, 0.7]$	0.50	0.50	0.50
$A_4 = [0.4, 0.8]$	0.63	0.62	0.64
$A_5 = [0.5, 0.7]$	0.67	0.64	0.72
$A_6 = [0.8, 0.9]$	1	1	1

Table 5.1 Comparison of seven intervals A_i against a reference one $B = [b_l, b_r] = [0.3, 0.7]$. The evaluations show the value of $P(A \geq B)$ with neutral, optimistic and pessimistic DM's attitudes.

Table 5.1 shows seven cases of comparison between intervals according to the previous definition of possibility degree and three different attitudes. For each interval $A = \{A_0, A_1, A_2, A_3, A_4, A_5, A_6\}$ the possibility degree of A being greater than $B = [0.3, 0.7]$, $P(A \geq B)$, is shown. An interval A_i has a possibility degree greater that 0.5 if the interval lies "above" B, and less than 0.5 if it lies "below". Note that, with all attitudes, the interval $A_0 = [0.1, 0.2]$ has zero possibility of being greater than interval $B = [0.3, 0.7]$ (it has no overlapping and is always lower than B) and the interval $A_6 = [0.8, 0.9]$ has a possibility of 1 (it has no overlapping and is always greater than B). All attitudes also lead to $P(A_3 \geq B) = 0.5$ because an interval always has a possibility degree of 0.5 of being greater than itself.

In what follows, and for the sake of clarity and simplicity, the neutral attitude detailed in Eq. (5.3) will be used.

5.2 Model

In this section an approach to identify a set of "Solutions of Interest" (SOIs) from a possibly large Pareto Front is described.

The general idea is to assign an interval to every solution. The intervals represent the range of potential scores that a solution can attain after an aggregation process. Then, using the possibility degree formulae, a comparison between those intervals is made which is later used to filter them obtaining the set of SOIs. The overall approach is shown in Fig. 5.1. There are four steps, namely, 1) intervals calculation, 2) reference interval identification, 3) intervals comparison and 4) set of SOIs calculation. These steps are explained below.

5.2.1 Step 1: intervals calculation

After solving a given MOP by any means, the result is a set of solutions $s_j \in \mathscr{P}^*$, $j = 1, \ldots, m$ that can be organised as a table where every row is a solution $\{s_1, s_2, \ldots, s_m\}$ and every column represents each objective function $\{f_1, f_2, \ldots, f_n\}$. Denoting $f_{ij} = f_i(s_j)$, the value of the solution s_j under the objective function f_i, without loss of generality, it is assumed that $f_{ij} \in [0,1]$, $i = 1, \ldots, n$, $j = 1, \ldots, m$.

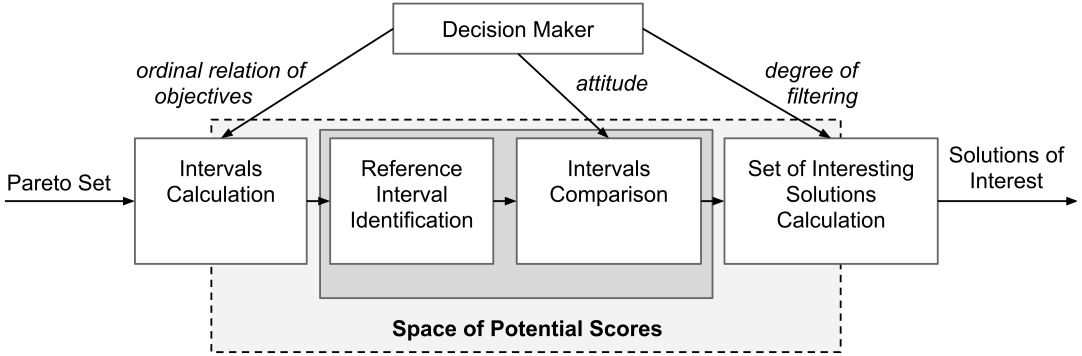


Fig. 5.1 Workflow diagram. The DM indicates the ordinal relation of objectives to calculate the intervals. Then, a reference interval is identified and the comparison between intervals is made according to the DM's attitude, obtaining an interval's evaluation. The last step is to use these evaluations to calculate the SOIs.

One simple way to detect SOIs is to assign a score to every solution and then rank them, keeping the top ones. This can be readily done using aggregation functions. An aggregation function $ag(F(s_j),W)$, $ag : [0,1]^n \rightarrow [0,1]$ combines the inputs (f_{ij}) into a single value (score) that can be later used to sort the solutions. There is a great diversity of aggregation functions. The reader may be referred to [5] for a complete introduction to the topic. One well known aggregation function is the weighted sum:

$$ag(F(s_j),W) = \sum_{i=1}^{n} w_i f_{ij}, \tag{5.6}$$

where $W = \{w_1, w_2, \ldots, w_n\}$ with $\sum w_i = 1$ and $w_i \in [0,1]$.

Regarding W, it should be noted that:

a) weights' determination is far from trivial and studies have shown that the use of aggregation affects the quality of the decision [46], and

b) any specific selection of the set of weights W leads to a different score for every solution.

Possible sets of weights could be obtained using *Surrogate weighting methods* [1] with the most common ones being the Rank Sum (RS) weights, Rank Reciprocal (RR) weights [79], Rank Order Centroid (ROC) weights and Equal (EW) weights [4] or more sophisticated methods [99, 100] in the event of uncertain linguistic environments.

In this proposal, it is assumed that the DM's preferences are given using an ordinal relation among the objectives (thus the DM does not need to provide a specific set of weights). It is assumed, without loss of generality, that this relation is $f_1 \succeq_p f_2 \succeq_p \cdots \succeq_p f_n$. Symbol $\succeq_p$ should be read as "is at least as preferred to". In terms of the weights, this implies $w_1 \geq w_2 \geq \cdots \geq w_n$.

The Potential Scores space is then defined as a bi-dimensional space that represents all the scores a solution s_j can obtain under the assumption of $w_1 \geq w_2 \geq \cdots \geq w_n$. It is built combining the inputs $f_{ij}, i = 1, \ldots, n$, into two values $I_j = [L_j, U_j]$ (the minimum and maximum scores that a solution s_j can attain) with $L_j \leq U_j$ and $L_j, U_j \in [0,1]$. It holds that $ag(F(s_j), W) \in [L_j, U_j]$, $\forall W$ satisfying $w_1 \geq w_2 \geq \cdots \geq w_n$, $\sum w_i = 1$ and $w_i \in [0,1]$.

Now the question is how such interval bounds L_j and U_j could be obtained. The solution of the two following independent Linear Programming (LP) problems, where $w_i, i = 1, \ldots, n$ are the decision variables, is the answer.

$$
\begin{aligned}
MIN \quad & L_j = \textstyle\sum_{i=1}^{n} w_i f_{ij}, \\
s.t. \quad & \\
& w_1 \geq w_2 \geq \cdots \geq w_n, \\
& \textstyle\sum_{i=1}^{n} w_i = 1, \\
& w_i \in [0,1].
\end{aligned}
\tag{5.7}
$$

$$
\begin{aligned}
MAX \quad & U_j = \textstyle\sum_{i=1}^{n} w_i f_{ij}, \\
s.t. \quad & \\
& w_1 \geq w_2 \geq \cdots \geq w_n, \\
& \textstyle\sum_{i=1}^{n} w_i = 1, \\
& w_i \in [0,1].
\end{aligned}
\tag{5.8}
$$

Nowadays, solving these linear programs is easy and very fast from a computational point of view. However, as stated in [1], there is no need to run an algorithm like Simplex to solve these specific LP problems, because "*by the well-known properties of a linear program, only the extreme points of the ranked weights need to be considered to effect the desired optimum and they are readily available*". This implies that, instead of solving the LP problems, solutions need to be scored only at the extreme points (three sets of weights). Then, the minimum and maximum are selected as L_j and U_j and the process is finished. Let n be the number of objectives, thus, the sets of weights to consider are:

1. $W = (1,0,\ldots,0)$: all the weight is assigned to the most preferred objective.

2. $W = (1/(n-1),1/(n-1),\ldots,1/(n-1),0)$: the least preferred objective is assigned $w_n = 0$ while the others get $w_i = 1/(n-1), \forall i \neq n$.

3. $W = (1/n, 1/n, \ldots, 1/n)$: all the objectives are equally important.

For example, having a decision problem with three objectives, $n = 3$, and $f_1 \succeq_p f_2 \succeq_p f_3$, every solution needs to be evaluated on just these three sets of weights: $(1,0,0)$, $(1/2,1/2,0)$ and $(1/3,1/3,1/3)$.

It should be noted that this space is always bi-dimensional regardless of the number of objectives the initial MOP has. In other words, the approach can be readily used in a many objective optimisation problem.

5.2.2 Steps 2 and 3: reference interval identification and intervals comparison

The next step is to compare the solutions s_j using their corresponding intervals in the Potential Scores space. Instead of using a full pairwise comparison, which would be computationally expensive, a reference solution s^* will be identified. The intervals $I_j = [L_j, U_j]$ will be compared against $I^* = [L^*, U^*]$ (the interval corresponding to s^*). The interval comparison method used is the one described in Sec. 5.1.3.

The reference solution s^* is the one with greatest lower bound value among all intervals in the Potential Scores space[2], i.e., the one which has $I^* = [L^*, U^*] \,|\, L^* \geq L_j, \forall\, j$.

[2]In the special case where this interval is not unique, the reference interval is the one that also has the greatest upper bound.

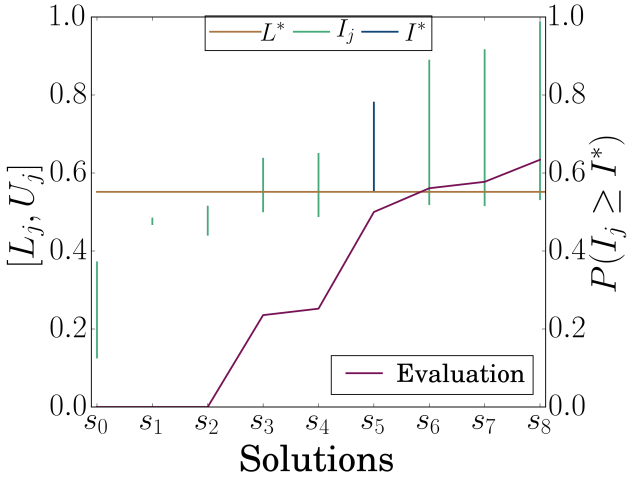


Fig. 5.2 Example of interval representation $I_j = [L_j, U_j]$ for each solution s_j. The reference interval is $I^* = I_5$. The evaluation line represents the value $P(I_j \geq I^*)$.

Then, for every solution s_j, the possibility degree $P(I_j \geq I^*) \in [0,1)$ is calculated following Eq. (5.2) (considering a neutral attitude, $f(x) = c$). As s^* is the solution with the greatest lower bound there is no interval that verifies $P(I_j \geq I^*) = 1$. This means that there is no solution that *always* scores better than the chosen reference solution. The solutions s_j such that their intervals I_j verify $P(I_j \geq I^*) = 0$ will always attain a lower score than s^*: there is no set of weights that can make s_j better than s^*. Also, the reference solution s^* will always score 0.5 as $P(I^* \geq I^*) = 0.5$.

Figure 5.2 shows the evaluation of $P(I_j \geq I^*)$ where $I^* = I_5$ (the reference solution is identified as $s^* = s_5$). This should be understood as follows: the possibility degree of interval $I_8 = [0.53, 0.99]$ being greater than the reference interval $I^* = [0.55, 0.78]$ is calculated as $P(I_8 \geq I^*) = \frac{0.99-0.55}{0.99-0.53+0.78-0.55} = 0.64$. In this example, the solutions s_0, s_1, s_2 will never score better than the reference solution $s^* = s_5$ because their intervals on the Potential Scores space have a possibility degree of zero of being greater than the reference interval, that is, $P(\{I_0, I_1, I_2\} \geq I^*) = 0$. In general, every interval that lies completely below the horizontal line (which represents L^*) has always zero possibilities of being greater than the reference interval and, because of that, the corresponding solution is always worse than the reference solution.

5.2.3 Step 4: set of solutions of interest calculation

Up to this point, every solution $s_j \in \mathscr{P}^*$ has an associated interval $I_j = [L_j, U_j]$ and the reference solution s^* is identified. The idea now is to filter out solutions to reduce the amount of information the DM has to evaluate.

The set of SOIs is defined as:

$$S_\lambda = \{s_j \mid P(I_j \geq I^*) > \lambda\}, \tag{5.9}$$

where $\lambda \in [0,1)$ is a value selected by the DM and stands for the minimum value of P that a solution needs to have in order to be considered an "interesting" solution. It holds that if $\lambda_1 < \lambda_2$ then $S_{\lambda_2} \subseteq S_{\lambda_1}$, the higher the value of λ, the smaller the size of S_λ. Also, $s^* \in S_\lambda \iff \lambda < 0.5$, the reference solution is only included on the set of SOIs when λ is less than 0.5 because the reference solution s^* will always score 0.5.

Some examples on SOIs sets obtained by using the solutions from Fig. 5.2 are given next. If $\lambda = 0$, the set of SOIs contains the solutions s_j with $P(I_j \geq I^*) > 0$. Then, $S_{\lambda=0} = \{s_3, s_4, s_5, s_6, s_7, s_8\}$, six solutions have more than 0 possibility of being greater than the reference. If $\lambda = 0.3$, the set of SOIs is $S_{\lambda=0.3} = \{s_5, s_6, s_7, s_8\}$, four solutions have a possibility degree higher than 0.3.

Once the set of SOIs is defined, the DM has a much smaller set of relevant solutions to work with.

5.3 Experimentation and results

In this section, the behaviour of the proposal on optimisation problems with two, three and many objectives is illustrated.

Initially, the DM needs to indicate the preferences among the objectives. Without loss of generality, it is assumed $f_1 \succeq_p f_2 \succeq_p \cdots \succeq_p f_n$ and $f_i \in [0,1]$, $\forall i$. For the following examples it is considered that the DM has a neutral attitude, see Eq. (5.3).

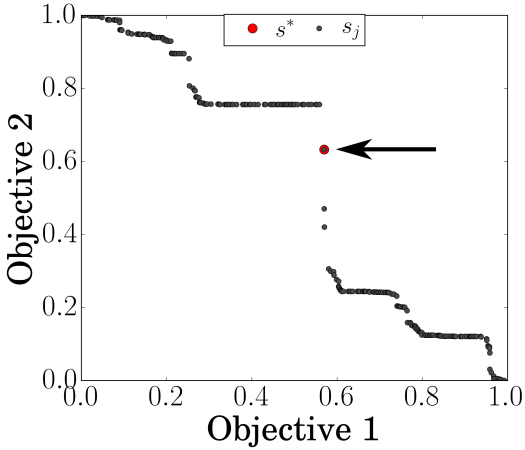


Fig. 5.3 Example A: initial Pareto Front.

5.3.1 Example A: two objectives with random pareto front

The objective of this first example is to provide a clear view of the sets of SOIs and their sizes obtained from the Pareto Front, shown in Fig. 5.3. It contains 300 solutions generated using a randomised greedy algorithm available at github.com/TorresM. The solutions are evaluated on two objectives with the objective 1 being preferred to the objective 2. Both of them are normalised.

Initially, the intervals on the Potential Scores space are obtained and the reference interval I^* is identified. In Fig. 5.3 the corresponding reference solution is marked with an arrow. Next, the possibility degree values $P(I_j \geq I^*)$ are calculated. Figure

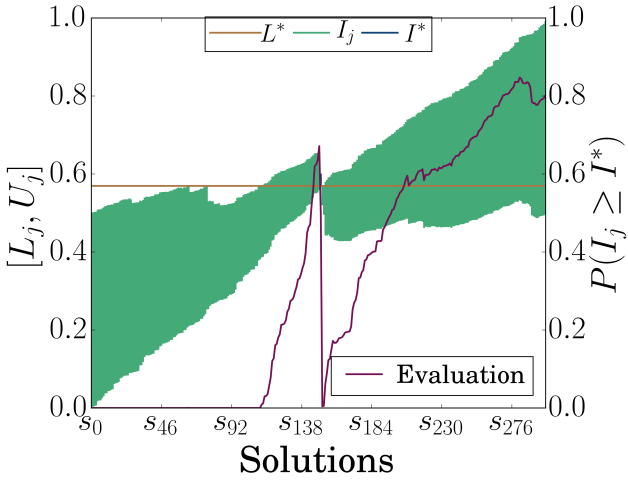


Fig. 5.4 Example A: the interval representation of I_j, $I^* = [L^*, U^*]$ and the evaluations $P(I_j \geq I^*)$ for every solution s_j. Solutions follow the x axis order from Fig. 5.3

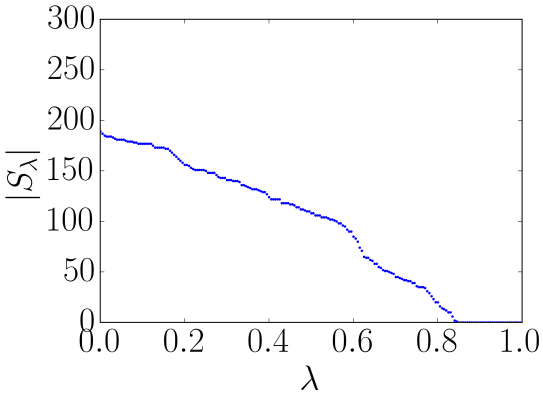


Fig. 5.5 Example A: value of $|S_\lambda|$ as a function of λ.

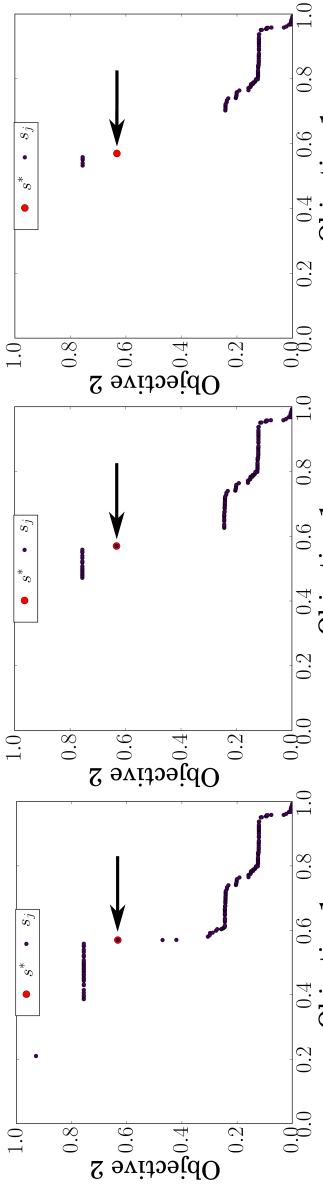


Fig. 5.6 Example A: sets (a) $S_{\lambda=0}$, (b) $S_{\lambda=0.25}$ and (c) $S_{\lambda=0.5}$. The reference solution s^* is identified with an arrow.

5.4 shows both the scores' intervals I_j and the possibility degree $P(I_j \geq I^*)$ for every solution, s_j. For this example, the reference interval corresponds to the solution s_{151}, $I^* = I_{151} = [0.57, 0.60]$.

Figure 5.5 shows the variation of $|S_\lambda|$ as a function of λ. When $\lambda = 0$, $|S_{\lambda=0}| = 189$, which, in turn, means that 111 solutions are not in the set of SOIs $S_{\lambda=0}$. In other words, 1/3 of the initial solutions are discarded. As λ increases, the size of the S_λ decreases. For example, $|S_{\lambda=0.25}| = 150$, $|S_{\lambda=0.5}| = 108$, $|S_{\lambda=0.6}| = 85$ and $|S_{\lambda=0.8}| = 20$. In this particular example, there is a clear linear relation between λ and $|S_\lambda|$.

Now the question is where these SOIs are located in the Pareto Front. It should be recalled that $f_1 \succeq_p f_2$. Figure 5.6 shows the sets (a) $S_{\lambda=0}$, (b) $S_{\lambda=0.25}$ and (c) $S_{\lambda=0.5}$. The location of the reference solution is also highlighted. Although $s^* \notin S_{\lambda=0.5}$, it is included in the plot for visualisation purposes.

5.3.2 Example B: three objectives problem

In this second example, taken from [12], the approach is applied to a larger Pareto Front corresponding to a three objective optimisation problem. The Pareto Front has 1612 solutions and is depicted in Fig. 5.7.

Firstly, the intervals on the Potential Scores space are calculated, considering $f_1 \succeq_p f_2 \succeq_p f_3$, and the reference interval is identified as $I^* = I_{1391} = [0.41, 0.56]$. The corresponding ref-

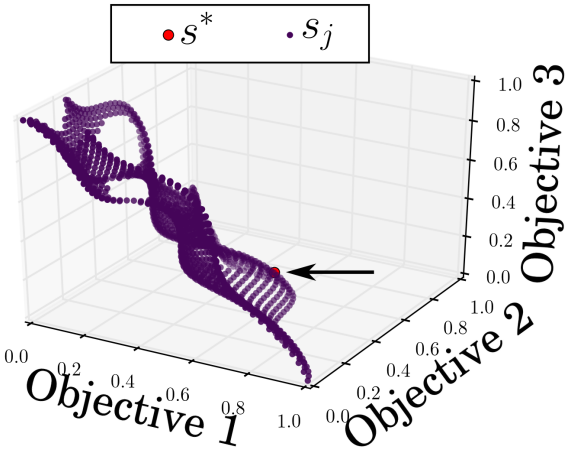


Fig. 5.7 Example B: initial Pareto Front.

erence solution is marked with an arrow. Then, the possibility degree of each interval $P(I_j \geq I^*)$ is calculated.

The intervals and their evaluations are shown in Fig. 5.8. In order to facilitate visualisation, the solutions are ordered by the possibility degree value.

Figure 5.9 shows the sizes of SOIs' sets for different λ values. In the case of $\lambda = 0$ the size of the set can be halved, $|S_{\lambda=0}| = 879$. If λ is further increased, smaller sets of SOIs can be achieved. For example, $|S_{\lambda=0.35}| = 382$ and $|S_{\lambda=0.7}| = 23$.

It is also relevant to know the location of the SOIs in the original Pareto Front. The sets (a) $S_{\lambda=0}$, (b) $S_{\lambda=0.35}$ and (c)

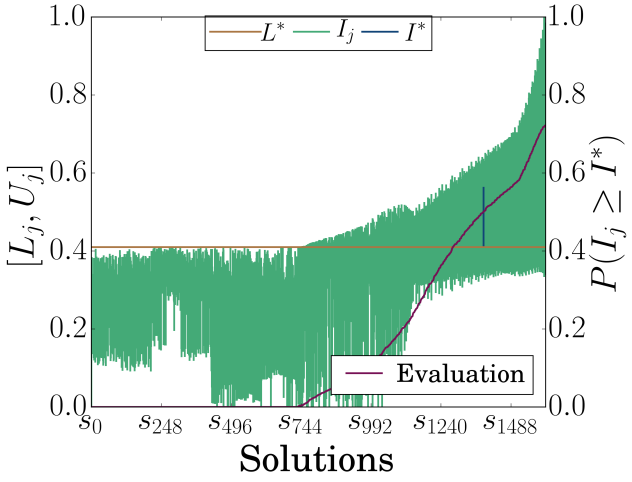


Fig. 5.8 Example B: interval representation I_j, $I^* = [L^*, U^*]$ and $P(I_j \geq I^*)$ for every solution s_j.

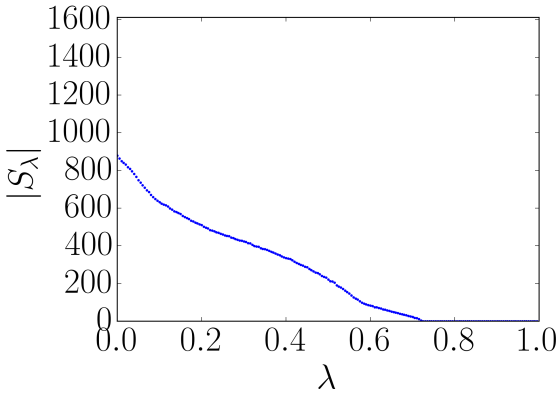


Fig. 5.9 Example B: value of $|S_\lambda|$ as a function of λ.

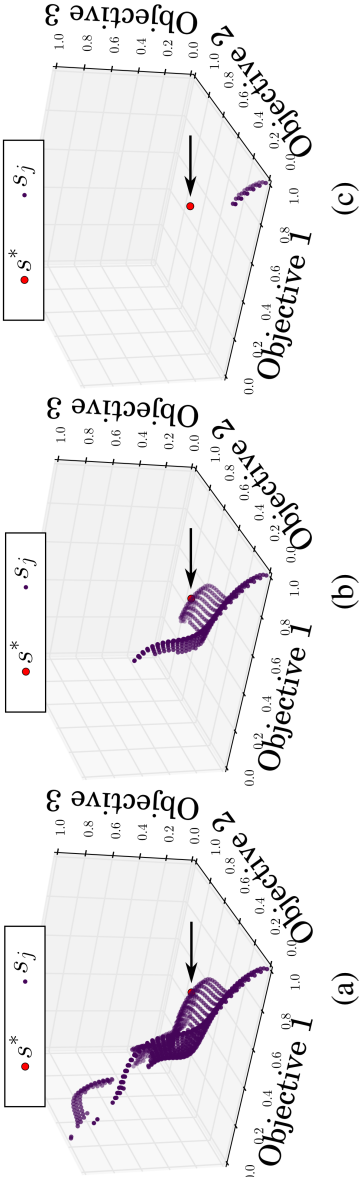


Fig. 5.10 Example B: sets (a) $S_{\lambda=0}$, (b) $S_{\lambda=0.35}$ and (c) $S_{\lambda=0.7}$. The reference solution s^* is identified with an arrow.

$S_{\lambda=0.7}$ are depicted in Fig. 5.10. Although $s^* \notin S_{\lambda=0.7}$, it is included in the plot for visualisation purposes.

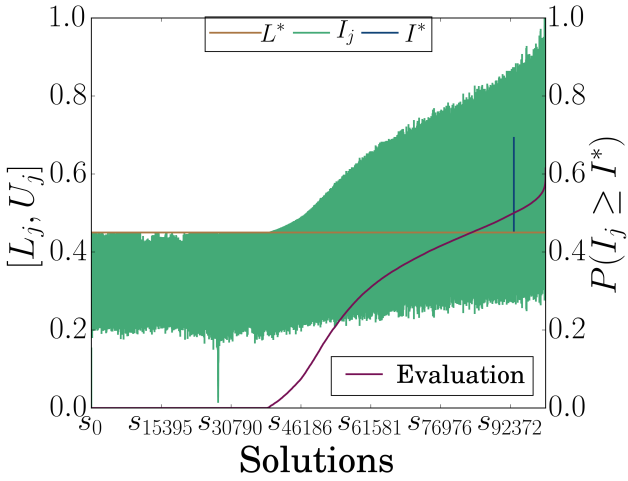
5.3.3 Example C: many objectives problem

The proposed approach is next applied to a real data case taken from [7]. The Pareto Front consists of 100070 non-dominated solutions from a 9 objective radar waveform design problem originally published in [37].

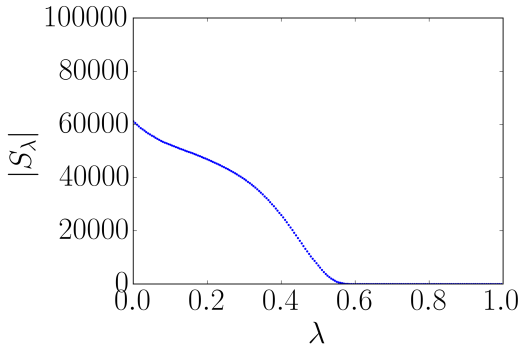
Following the same steps as before, the intervals on the Potential Scores space are obtained, the reference interval $I^* = I_{93085} = [0.45, 0.69]$ is identified and the possibility degrees $P(I_j \geq I^*)$ are calculated (elements depicted in Fig. 5.11 (a)). In order to facilitate the visualisation, the solutions are ordered by their possibility degree value.

The relation between $|S_\lambda|$ and λ is shown in Fig. 5.11 (b). Departing from a set of 100070 solutions, and considering the following sets of SOIs: $|S_{\lambda=0}| = 61271$, $|S_{\lambda=0.2}| = 46949$, $|S_{\lambda=0.4}| = 25983$ and $|S_{\lambda=0.6}| = 3$, reductions of $39\%, 54\%, 74\%$ and 99.9% are obtained, respectively.

The solutions in $S_{\lambda=0.6}$ are displayed in Fig. 5.12. Although $s^* \notin S_{\lambda=0.6}$, it is included in the plot for visualisation purposes.



(a)



(b)

Fig. 5.11 Example C: (a) interval representation and evaluation of $P(I_j \geq I^*)$ for every solution s_j. In (b), the value of $|S_\lambda|$ as a function of λ.

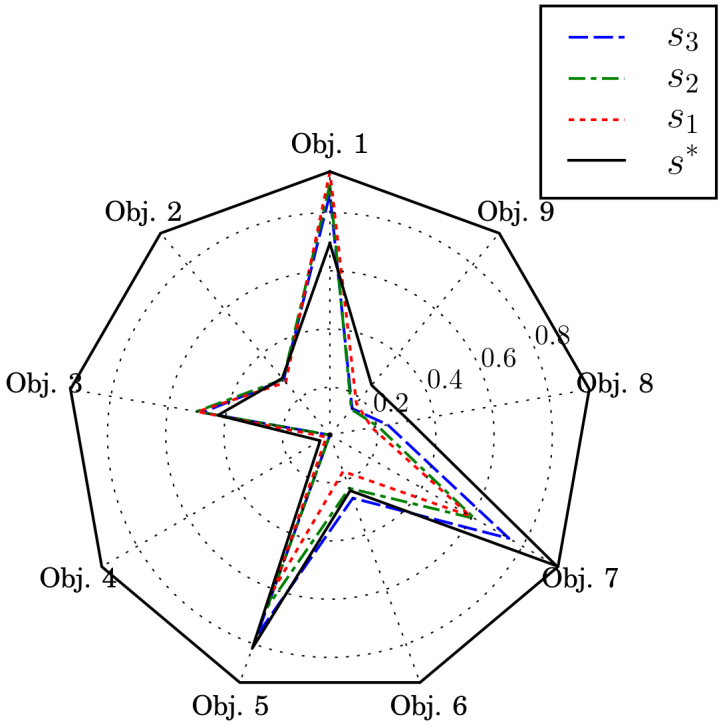


Fig. 5.12 Example C: set $S_{\lambda=0.6}$. The reference solution s^* is included in the plot for visualisation purposes.

Algorithm 4 Pseudocode of the data generation

```
for i=1 to 200 step 1 do
    pareto ← randomly obtain 100 non-dominated solutions on 3 objectives
    intervals ← calculateIntervals(pareto)
    I* ← identifyReference(intervals)
    for j=1 to 100 do
        calculate P(I_j ≥ I*) with neutral attitude
    end for
    for λ=0 to 1 step 0.05 do
        S_λ ← solutions j with P(I_j ≥ I*) > λ
    end for
end for
```

5.3.4 On the relation between λ, $|S_\lambda|$ and $|S|$

The previous examples showed the role of the parameter λ as a filtering degree. The ability to reduce the set of solutions in S_λ can be further analysed exploring the relation $|S_\lambda|/|S|$ where $|S|$ is the number of initial solutions on the Pareto Front. For that purpose, the procedure shown in Algorithm 4 is used.

First, it creates a set S of 100 non-dominated solutions s with 3 objectives using a randomised iterative greedy algorithm to generate a pareto set available online at github.com/TorresM. For each solution s_j, the interval of scores I_j on the Potential Scores space is calculated. Then, the reference interval I^* is identified and the possibility degrees $P(I_j \geq I^*)$ are calculated according to a neutral attitude of the DM. Finally, the sets S_λ

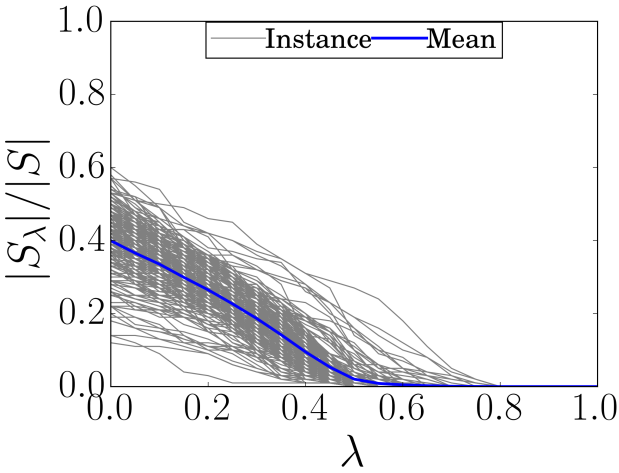


Fig. 5.13 Reduction indexes obtained from experimentation with 200 instances of 100 non-dominated solutions sets.

with $\lambda = \{0, 0.05, 0.1, ..., 0.95, 1\}$ are obtained. The process is repeated 200 times.

The results are shown in Fig. 5.13. The first element to highlight is that, by setting $\lambda = 0$, the SOIs sets exclude only those solutions that would never be interesting for the DM because they will never attain a score better than the one of the reference solution. Such solutions can be readily discarded with no "risk". If solutions s_j such that $P(I_j \geq I^*) = 0$ are eliminated, on average $|S_{\lambda=0}|/|S| = 0.4$. That is, the size of the set is reduced by 60%. Also, $|S_{\lambda=0.4}|/|S| = 0.1$ which means that the original set is reduced on average by 90%.

For previous examples, A, B and C, $S_{\lambda=0}$ sets are considerably reduced: $|S_{\lambda=0}|/|S| = 0.63$, $|S_{\lambda=0}|/|S| = 0.55$ and $|S_{\lambda=0}|/|S| = 0.61$, respectively. This means that even when just those solutions s_j with $P(I_j \geq I^*) = 0$ are eliminated, the reductions are around 40% of the initial Pareto Front sets. Smaller sets of SOIs can be obtained with higher λ values, like $|S_{\lambda=0.8}|/|S| = 0.07$, $|S_{\lambda=0.7}|/|S| = 0.01$ and $|S_{\lambda=0.6}|/|S| = 0.01$ for examples A, B and C, respectively.

In its current implementation, the approach is not able to directly identify a set of SOIs with a specific number of solutions. Nevertheless, as is clear from previous paragraphs, the DM can increase or decrease the λ value to find smaller or larger sets of SOIs.

Finally, the approach has been shown to work with large sets of non-dominated solutions and many-objectives as easily as with smaller problems. This feature is possible due to the use of the space of Potential Scores for the intervals comparison.

5.4 Example of application: Strain design

The proposed approach was applied on Synthetic Biology, an area of maximum relevance nowadays. This application was possible thanks to a collaboration with Newcastle's Interdisciplinary Computing and Complex BioSystems (ICOS) research group.

Microorganisms are a very promising platform for the production of industrially relevant biochemicals, as they can be used to obtain antibiotics, amino acids or biomaterials. Whatever the desired product is, it is important to have a good understanding of metabolic networks of the host. This network can be manipulated such that the flux can be directed toward the desired "target" product without rendering the cell inviable.

With recent advances in genome sequencing technologies and high-throughput screening experiments, computational biologists are able to reconstruct genome-scale metabolic networks with high prediction accuracy for a range of well-characterised microorganisms, such as *E. coli* [27, 65] and *B. subtilis* [34]. This is particularly beneficial for strain design as high-quality metabolic networks allow to predict phenotypes from genotypes in an efficient manner and it is aimed at the identification of genetic perturbations needed for the production of biochemicals of interest.

A metabolic network of an organism can be represented as a flow network, see Fig. 5.14 (a), where the metabolites are represented as nodes in the network and the arcs are the reactions between metabolites on that metabolic network. Each arc has a capacity that cannot be exceed by the flow/flux that goes through that arc. Strain design aims to maximise two primary objectives, i.e. the bioengineering objective (the target product) and the cellular objective (mainly biomass), which are often in conflict. A flux distribution associated to the reactions will determine the

production of the target product (the quantity of flux that can be directed toward the product).

Figure 5.14 (a) illustrates a very simple network where fluxes $\bar{v}_{uptake}$, $\bar{v}_{biomass}$ and $\bar{v}_{target}$ represent the uptake of metabolite M1 from the surrounding media, the formation of biomass, and the production of the chemical of interest (the target), respectively. Multiple flux distributions can be obtained within that network. For example, all uptake flux goes throw reaction $M1 \rightarrow M2$ leading to $\bar{v}_{biomass}$ equal to $\bar{v}_{uptake}$, as illustrates 5.14 (b). In that case the chemical of interest will not produce ($\bar{v}_{target} = 0$). A second possibility is that all flux goes throw reaction $M1 \rightarrow M3$. In that case, both $\bar{v}_{biomass}$ and $\bar{v}_{target}$ could be equal to $0.5\bar{v}_{uptake}$ as illustrated in Fig. 5.14 (c). However, if all flux arriving to M4 goes to reaction $M4 \rightarrow M5$, the chemical target will have no production ($\bar{v}_{target} = 0$), as illustrated in Fig. 5.14 (d). Those three cases are not the only possible flux distributions as the fluxes arriving to a metabolite can be distributed following different reactions in different ratios. As seen, different distributions could lead to the same production rate and, also, when a production rate is calculated it usually consists on an interval with minimum and maximum achievable values of production, which can sometimes match. For example, the production rates for the network shown in 5.14 (a) are the intervals $\bar{v}_{biomass} = [0, \bar{v}_{uptake}]$ and $\bar{v}_{target} = [0, 0.5\bar{v}_{uptake}]$ due to all possible flux distributions detailed before.

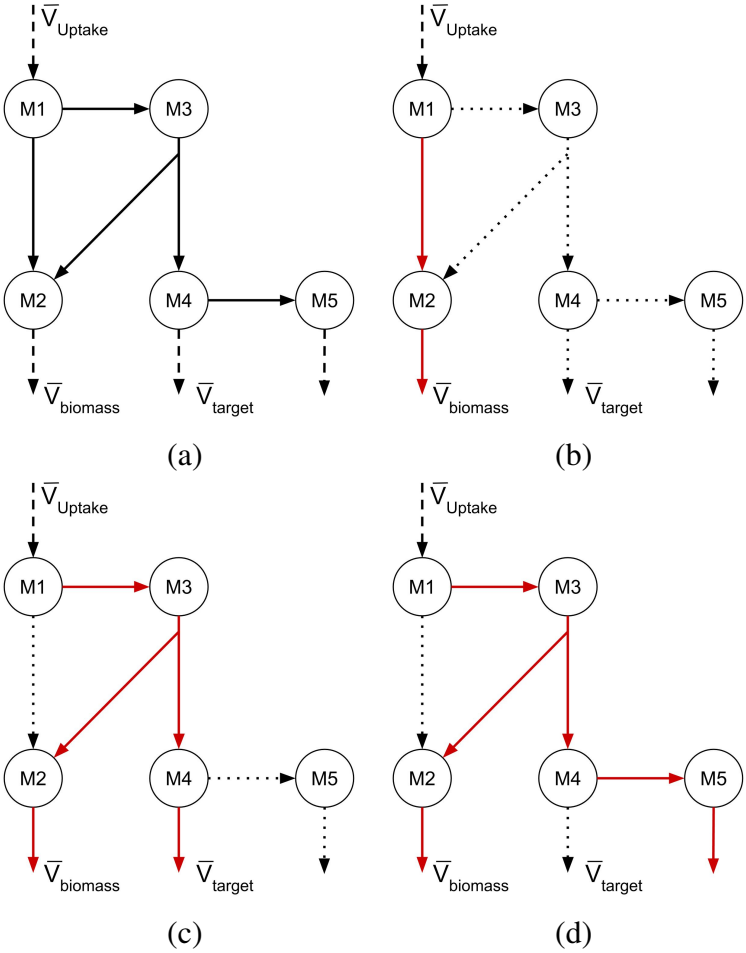


Fig. 5.14 Metabolic network illustration: (a) simple graph with 5 metabolites, (b) flux distribution leading to $\bar{v}_{biomass} = \bar{v}_{uptake}$, (c) flux distribution leading to $\bar{v}_{biomass} = \bar{v}_{target} = 0.5\bar{v}_{uptake}$ and (d) flux distribution leading to $\bar{v}_{biomass} = 0.5\bar{v}_{uptake}$ and $\bar{v}_{target} = 0$. Example taken from [82].

An ideal situation for the metabolic network in question is the one depicted in Fig. 5.14 (c). It maximises both the production of the chemical of interest ($\bar{v}_{target} = 0.5\bar{v}_{uptake}$) and the formation of biomass ($\bar{v}_{biomass} = 0.5\bar{v}_{uptake}$). To force that specific flux distribution two knockouts are need to be performed: on reactions $M1 \rightarrow M2$ and $M4 \rightarrow M5$. That way, flux distributions depicted in (b) and (d) are no longer possible.

The reactions of a network are determined by the genes of the organism. The gene-protein-reaction (*GPR*) relationship describes the links between gene activities and reactions. Fig. 5.15 presents four common gene-protein-reaction associations. The removal of a gene gives rise to the loss of the gene product, i.e., enzymatic proteins, and thus leads to the deactivation of the gene-encoding reactions, that is, the reactions are no longer active in the network. As a consequence, there exists no flux for the reactions associated with the removed genes. The idea in gene-knockout based metabolic engineering then, is that the target production (the bioengineering objective) can be increased by removing genes that catalyse reactions on competing pathways.

The strain design method used for this example is a multi-objective variable neighbourhood search (MOVNS). It focuses on the removal (knockouts) of reactions from a reconstructed metabolic network of organisms to increase target product formation. The resulting gene deletions can be computed using the gene-protein-reaction relationship. As the sizes of typical metabolic network models are around 1800 metabolites, 2500

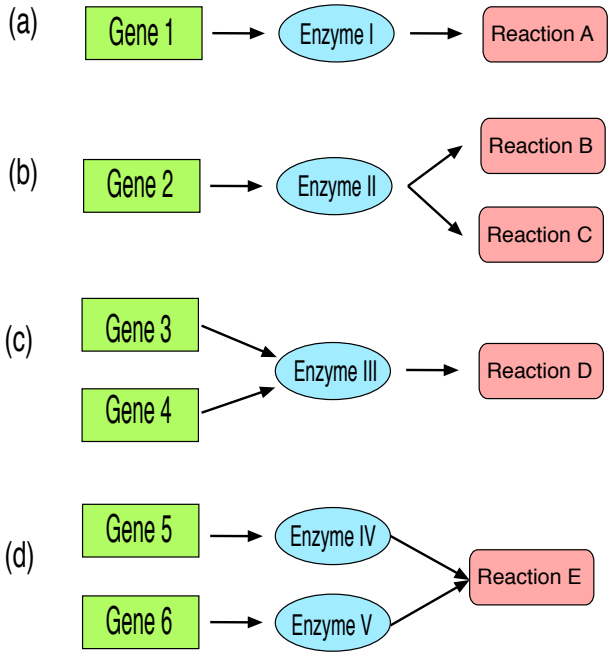


Fig. 5.15 The gene-protein-reaction relationship in four scenarios. (a) gene 1 has a unique function for reaction A. (b) gene 2 has multifunctions, catalysing both reaction B and C. (c) gene 3 and gene 4 encode an enzyme complex to catalyse the enzymatic reaction D. (d) gene 5 and gene 6 are isozyme genes, each of which can catalyse reaction E.

reactions and 1300 genes it is common to obtain a very large set of solutions. Having such set of solutions the identification of SOIs simplifies later analysis.

5.4.1 Identify SOIs applied to strain design

In next examples *E. coli* metabolic networks are redesign for the production of two commercially valuable chemicals, i.e., succinic acid and naringenin. Succinic acid is a native metabolite in *E. coli* whereas naringenin is nonnative and a heterologous pathway (a subgraph of metabolites and reactions derived from a different organism) needs to be added to the *E. coli* metabolic network. For succinic acid, the *E. coli* core model [66] is used, while the iAF1260b model [65] is used for naringenin production.

Four criteria are considered to select and identify SOIs from the multiobjective optimisation problem's solutions:

c_1: $v_{biomass}$, the maximum flux toward biomass,

c_2: v^b_{target}, the maximum target production,

c_3: v^w_{target}, the minimum target production, and

c_4: number of actual knockouts (limited by K).

The set of initial solutions are obtained from a multiobjective optimisation problem that maximises $v_{biomass}$, v^b_{target} and v^w_{target}. Then, the approach to identify SOIs is applied to obtain the $|S_{\lambda=0}|$

sets according to the considered preferences on the criteria. It is assumed that the DM only want to discard solutions with $P = 0$. The *SOIs* set can be partially ordered as a ranking according to their evaluation.

Results on succinic acid

In this example different number of removals are considered, i.e. different K values. Thus, the *E. coli* core model is considered for succinic acid overproduction while K increases from 2 to 10.

Two different order relations are chosen by the DMs: $order_1 := c_3 \succeq_p c_2 \succeq_p c_1$ and $order_2 := c_2 \succeq_p c_3 \succeq_p c_1$. Using those orders two different sets of SOI_i will be obtained: when the criterion c_3 (the v^w_{target}) is the preferred one and when the criterion c_2 (the v^b_{target}) is the preferred one. The criteria c_4 (number of actual knockouts) is not considered for this example.

Table 5.2 shows the number of initial solutions, the size of SOI_i and, for the sake of simplicity, only the best ranked solutions (the ones with higher evaluation P), $best_i$, in each SOI_i. As seen from the table, the more knockouts allowed, the more solutions and SOIs obtained. The procedure incorporating the DM's preference reduces the solutions significantly. Particularly, when K=2 or 3, only a few solutions could be found to overproduce succinic acid, and they are also a part of best solutions for larger K values ($K \leq 5$).

Table 5.2 Results on succinic acid: number of solutions obtained with MOVNS for different K, the size of SOI_1 and SOI_2 and the best solutions $best_1$ and $best_2$. Reaction identifiers are consistent with *E. coli* core model [66].

K	$\|S\|$	$\|SOI_1\|$	$best_1$	$\|SOI_2\|$	$best_2$
2	3	1	CO2t PGI	1	CO2t PGI
3	50	8	CO2t PGI (ACKr or ACt2r)	8	CO2t PGI (ACKr or ACt2r)
4	135	35	CO2t PGI (PFL or FORt (GLNabc or AKGt2 or AKGDH or ACT2r or ACKr)	35	CO2t PGI (ACt2r or ACKr or (PTAr (SUCOAS or SUCDi)) or (ACt2r SUCDi) or (ACKr (FORt2 or FUMt2_2)))
5	234	53	CO2t PGI PYRt2 ((PFL THD2) or (FORt PYK))	54	CO2t PGI PYRt2 ((PFL THD2) or (FORt PYK))
6	384	57	CO2t PGI PYRt2 ((PFL THD2) or (FORt PYK))	56	CO2t PGI PYRt2 ((PFL THD2) or (FORt PYK))
7	466	59	CO2t PGI PYRt2 ((PFL THD2) or (FORt PYK))	58	CO2t PGI PYRt2 ((PFL THD2) or (FORt PYK))
8	561	71	CO2t PGI PYRt2 ((PFL THD2) or (FORt PYK))	68	CO2t PGI PYRt2 ((PFL THD2) or (FORt PYK))
9	598	97	CO2t PGI PYRt2 ((PFL THD2) or (FORt PYK))	67	ACALD ATPS4r D_LACt2 ME2 THD2 PTAr FORt2 SUCCt2_2 G6PDH2r
10	638	107	CO2t PGI PYRt2 ((PFL THD2) or (FORt PYK))	77	ACALD ATPS4r D_LACt2 ME2 THD2 PTAr FORt2 SUCCt2_2 ((FBP GPL) or G6PDH2r)

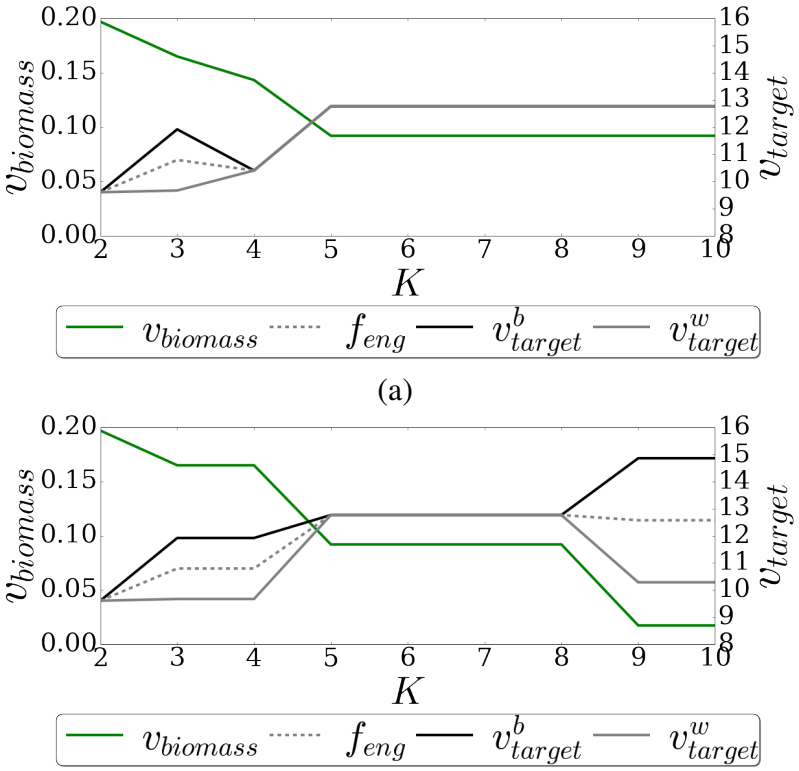


Fig. 5.16 Results on succinic acid: criteria values of the $best_i$ evaluated solution (in terms of P) for different values of K under (a) $order_1$, c_3 is the most important criterion, and (b) $order_2$, c_2 is the most important criterion.

The information provided as SOIs set allows a deeper analysis on the solutions. Fig. 5.16 shows the v_{biomss}, v^b_{target}, v^w_{target}, and $f_{eng} = 0.5v^w_{target} + 0.5v^b_{target}$ values of the best solution in each SOI_i when K increases from 2 to 10. It is clear that f_{eng} generally increases and v_{biomss} decreases with larger K values. The figure shows v^b_{target} is easier to be improved compared with v^w_{target}. It also indicates that the minimal target production levels out after $K = 5$. Thus, the selection of K is critical as it has to strike a balance between knockout costs (consisting of both the number of genetic removals and biomass loss) and target production. Besides, the difference between the two subplots in the figure clearly demonstrates that the proposed approach is able to filter solutions according to the order relation of criteria incorporating the DM's preferences. In (a) the v^w_{target} is preferred to v^b_{target} while in (b) is the other way around. For that reason, when $K = 4$ the best solution provided to each DM is different. This difference is also observed when $K = 9$ and $K = 10$.

Results on naringenin

Three hypothetical pathways, listed in Table 5.3, are added to the network for naringenin production. Addition of these pathways to iAF1260b [66] results in three expanded models, named M1, M2, and M3, respectively. Table 5.3 also presents the theoretical maximum growth rate (GR) and production rate (TMPR) computed by Flux Balance Analysis (FBA) (a mathematical method

Table 5.3 Three naringenin-production pathways with growth rates (GR: h^{-1}) and theoretic maximum production rates (TMPR: mmol $gDW^{-1}h^{-1}$).

No.	Pathway description	GR	TMPR
1	3-dehydroshikimate→3,4-dihydroxybenzoate→ 3,4-dihydroxybenzaldehyde→2',3,4,4',6'-pentahydroxychalcone→eriodictyol→naringenin	0.7814	3.6084
2	L-tyrosine→L-dopa→trans-caffeate→caffeoyl-coa →2',3,4,4',6'-pentahydroxychalcone→eriodictyol →naringenin	0.7814	3.3977
3	L-phenylalanine→L-tyrosine→L-dopa→trans-caffeate→caffeoyl-coa→2',3,4,4',6'-pentahydroxychalcone→eriodictyol→naringenin	0.7814	2.1287

for simulating metabolism in genome-scale reconstructions) for the models using different pathways. Judging from GR and TMPR values, pathway 1 is theoretically the most efficient one for naringenin production before reaction knockout, followed by 2 and then by 3.

To know which pathway is most efficient for naringenin production after reaction knockouts, all obtained solutions from the three models are considered as the initial solutions set. That is, the approach to identify SOIs is applied to all solutions obtained with $K = 5$ for the three expanded models considering two preference orders: $order_1 := c_3 \succeq_p c_2 \succeq_p c_1 \succeq_p c_4$ and $order_2 := c_2 \succeq_p c_3 \succeq_p c_1 \succeq_p c_4$, leading to SOI_1 and SOI_2.

The results obtained from the strain design framework are summarised in Table 5.4 where the number of initial solutions

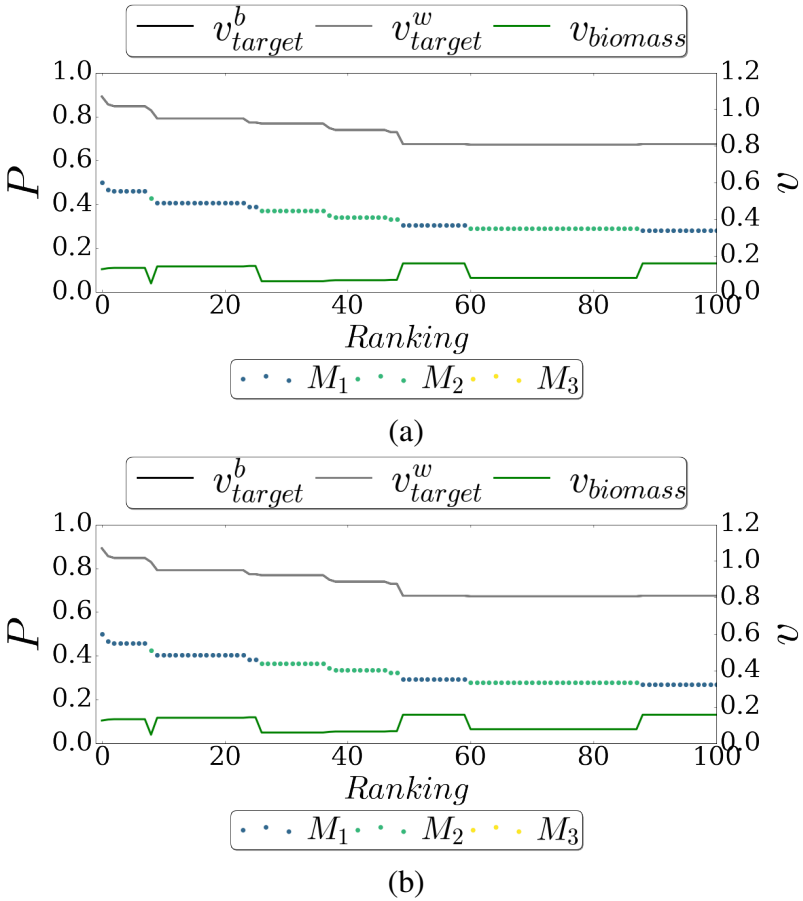


Fig. 5.17 Evaluation P and values of c_1, c_2 and c_3 for the top 100 solutions on the ranking considering models M_1, M_2 and M_3. The orders are (a) $order_1$, c_3 is the most important criterion, and (b) $order_2$, c_2 is the most important criterion.

Table 5.4 Results on naringenin: Number of solutions obtained with MOVNS for different models, the number of solutions in SOI_1 and SOI_2 and the best solutions $best_1$ and $best_2$. Reaction identifiers are consistent with iAF1260 [27].

$Model$	$s \in S$	$s \in SOI_1$	$P(best_1)$	$s \in SOI_2$	$P(best_2)$	$best_1 = best_2$
$M1$	690	263	0.5	263	0.5	GART MDH O2tpp PPC GHMT2r
$M2$	793	167	0.4266	162	0.4266	GART MDH O2tpp (PGCD or PSERT) PPC
$M3$	352	2	0.0269	2	0.0256	(PPNDH or PHET)

for each model, the number of solutions from that model that is also included in the set SOI_i and, for the sake of simplicity, only the best ranked solutions, $best_i$, (the ones with higher evaluation P) for each model are shown. In this example, the best solutions are different across three models but are the same under different preference orders.

The SOIs sets help to understand the quality of the solutions and to analyse the solutions. For example, in this specific application, it can be observed that M1, the model with the most efficient naringenin production pathway, has more solutions in the set SOIs compared with the other models, which means that many more knockout options can be used for high production of naringenin.

Fig. 5.17 depicts the P values, $v_{biomass}$, v^b_{target} and v^w_{target} for the top 100 solutions on that set. As seen, the top-ranking solutions are mainly from M1, confirming that the first pathway

is indeed the most efficient for naringenin production. The third pathway is the worst as no solution from M3 appears on the top 100 list and the best solution from M3 has a low P value.

It can be observed that two subplots are identical, that is, top 100 solutions in SOI_1 are the same as in SOI_2. The difference between both SOI_1 and SOI_2 is the preference order (first v^w_{target}, then v^b_{target} for SOI_1, and vice versa for SOI_2). However, in this application the v^b_{target} and v^w_{target} have the exact same value for each solution on the top 100, so the results are the same under both preference orders.

There are some cases where solutions achieving similar naringenin production have better $v_{biomass}$ than others but are assigned worse rankings (e.g. M2 solutions ranked at 60$\sim$90 have worst $v_{biomass}$ than M1 solutions ranked at 90$\sim$100). A further investigation on these special cases reveals that the design cost (number of actual knockouts, c_4 criterion) in the former solutions is smaller than that in the latter ones.

5.5 Conclusions

The consideration of the DM's preferences is a key aspect when dealing with multi or many objective optimisation problems (MOP). An *a posteriori* approach is proposed to help the DM to obtain a reduced set of solutions of interest (SOIs) after the MOP solving process is completed.

This approach, in contrast with others based on geometrical features, associates the concept of "interest" in terms of the DM's preferences. Moreover, the DM's attitude can be incorporated into the possibility distribution function used to compare intervals and a filtering degree (parameter λ) can be adjusted to control the size of the set of SOIs.

One of the main advantages of the method is that the number of objectives of the MOP is somehow irrelevant as the comparison of solutions is done over the corresponding intervals and, once the reference solution is identified, just $N-1$ (with N being the size of the Pareto Front) intervals comparisons are needed in order to obtain the initial set of SOIs.

Overall, the proposal is simple to understand and implement. Due to a theoretical result, the calculation of the intervals bounds does not require running an optimisation algorithm and they can be readily obtained. This is very important if the DM wants to change his/her preferences: after a new ordinal relation of the objectives is proposed, the intervals should be recalculated. As this process is fast and easy from a computational point of view, the DM may easily explore other regions of the Pareto Front.

The identification of SOIs is a very useful tool with multiple fields of application as seen with Synthetic Biology example. It is able to provide best competing pathways according to the DM's preferences and, in general, the provided SOIs set is easier to analyse than the whole set of initial solutions.

Chapter 6

Conclusions

This chapter details the conclusions and future work as well as possible extensions of the work developed in this research.

The main objective proposed for this work was the use of Soft Computing for the improvement and resolution of decision and optimisation models on the adaptive maps scenario. This objective was subdivided into three sub-objectives:

1. Define a model to solve the personalised route problem and provide a solution approach.

2. Define a model for adaptive maps and provide a solution approach.

3. Define a posteriori method to select routes by incorporating preferences after the optimisation process.

The first sub-objective is accomplished on Chapters 2 and 3, where the personalised route problem (PRP) has been defined and its model has been described considering crisp and fuzzy constraints. The model has been shown to be versatile as it can be used to represent multiple scenarios of the real wold as well as to be suitable for multiple users with different preferences.

The solution approach to solve the PRP has been detailed. Moreover, an Android App called PRoA was implemented and fully explained. Using real examples on Granada, the usefulness of the approach was demonstrated. In some examples, PRoA obtained better (shorter) routes than other routing applications.

The proposed model for PRP with fuzzy constraints allows the users to consider imprecision in the specifications of their preferences. In order to solve the problem, an approach based on α-cuts was proposed. This allows the user to deal with different degrees of satisfaction of the constraints, going from crisp constraints to constraints with a maximum tolerance. The more tolerance is allowed the more relaxed is the problem and, potentially, a better route in terms of cost could be found.

The idea of defining α-cuts is an generalised method consisting on choosing fixed α values at the beginning of the process. A different more adaptive approach is proposed on this work. It consists on choosing the degree of satisfaction of the constraints in an iterative way, instead of fixing every α-cut at the beginning of the process, each α-cut is chosen according to the solutions already known. The adaptive approach could be able to reduce

the amount of duplicated solutions and is capable of finding different and potentially more variate solutions when solving the same number of α-cuts.

Chapter 4 focuses on the second sub-objective: the adaptive maps. The adaptive maps were defined and a model was proposed. The solution, an adaptive map, consists on a map's image that has been modified to fit the user's preferences. A method to generate such modified image is also detailed. As examples, multiple adaptive maps are shown using real data.

Sub-objective three was addressed in Chapter 5, where an *a posteriori* approach is proposed to help the decision maker (DM) to obtain a reduced set of solutions of interest (SOIs) from a large set of solutions. In this work the concept of "interest" was in terms of the DM's preferences.

A parameter, $\lambda \in [0,1]$, indicated a filtering degree on the generation of the SOIs set. This parameter allows to obtain smaller sets when λ is higher. Using multiple examples, it was shown which parts of the Pareto Front were kept under different DM's preferences and filtering degree values (λ values).

The proposed method was able to deal with both Multiobjective and Many objective optimisation problems as it filters SOIs from the Pareto Front independently of the number of objectives. It could be observed that, in general, the size reduction for the SOIs set, with respect to the original Pareto Front, was significant even with low values of λ.

An example of application in Synthetic Biology showed that the method is versatile and could be a very useful tool with multiple fields of application. In that specific example, it was proved to be able to provide best competing pathways according to the DM's preferences and, in general, the provided SOIs sets were easier to analyse than the whole sets of initial solutions. The approach seems promising according to experts at Newcastle University's laboratories who showed interest on possible further analysis of the results.

6.1 Future work

Significant improvements and possible extensions for research developed in this work are detailed next.

Personalised route problem

The pedestrian scene definition can be improved. It could be interesting to explore possible metrics to assess how green is a route and other qualitative attributes like"safety" or "entretainment". Fuzzy numbers and linguistic expressions could be of use when modelling and managing those attributes. Despite their application has not been further studied and explored, they could be also feasible given the flexibility of the presented PRP model.

A second improvement in this topic is the consideration of dynamic attributes and scenes. For example, thinking of "safety",

after some years or even at different hours on the same day the "safety" may not be the same. The same idea can be also applied to more common attributes like travel time where the time need to complete a route may differ from summer to winter.

In addition, different scenes need to be described, like safe routes or cycling route. It can be achieved using different sets of criteria that need to be properly defined.

Finally, the combination of fuzzy constraints with fuzzy linguistic variables can be explored. If the linguistic variables are adjusted to the range of possible values on the zone for the criteria, the user does not need to know the characteristics of the map a priori in order to set a feasible constraint, but he/she can directly choose a linguistic variable instead.

Adaptive maps

Despite the quality of the final Adaptive Map images, there is still room for improvement at several stages. One is the calculation of the new locations of the points of interest. Right now, Multidimensional Scaling is used but other techniques like Distance Geometry [60] should be explored.

Another aspect is visualisation. The proposed triangulation and mapping methods result in good map images in practice but further comparison will be a matter of interest.

Note that the dissimilarity matrix on the Multidimensional Scaling problem solved is required to be symmetric. That means

that in case the subjective matrix is asymmetric the results are obtained taking into account only half of the costs. The method can be drastically improved, avoiding this issue, by solving an Asymmetric Multidimensional Scaling problem [107] instead.

Solutions of interest identification

Current proposal on solutions of interest (SOIs) identification departs from a large set of solutions. It could be interesting to use a set of SOIs obtained from a geometric perspectives (like knee solutions) as a starting point instead.

It could be interesting to apply multicriteria decision making methods, once the SOIs set is obtained, to rank the solutions using a more sophisticated modelling of the DM's preferences (like using pairwise comparison among solutions).

A major improvement could be using different aggregation functions for the intervals' calculation, which will lead to a different Potential Scores space.

Finally, a visualisation tool could be implemented with the objective of showing the relation between the solutions at the Potential Scores space with the initial set of SOIs. Such tool may help to better understand the relation among these spaces.

Chapter 7

Conclusiones

Este capítulo detalla las conclusiones y el trabajo futuro, así como las posibles extensiones del trabajo desarrollado en esta investigación.

El principal objetivo propuesto para este trabajo fue el uso de Soft Computing para la mejora y resolución de modelos de decisión y optimización en el escenario de mapas adaptativos. Este objetivo se subdividió en tres subobjetivos:

1. Definir un modelo para resolver el problema de la ruta personalizada y proporcionar un enfoque de solución.

2. Definir un modelo para mapas adaptativos y proporcionar un enfoque de solución.

3. Definir un método a posteriori para seleccionar rutas incorporando preferencias después del proceso de optimización.

El primer sub-objetivo se logra en los Capítulos 2 y 3, donde se ha definido el problema de ruta personalizada (PRP) y se ha descrito su modelo considerando restricciones "crisp" y restricciones difusas. Se ha mostrado que el modelo es versátil, ya que se puede usar para representar múltiples escenarios del mundo real, y es adecuado para múltiples usuarios con diferentes preferencias.

Se detalló el enfoque de solución para resolver el PRP. Además, se explicó por completo una aplicación de Android implementada llamada PRoA. Utilizando ejemplos reales sobre Granada, se demostró la utilidad del enfoque. En algunos ejemplos, PRoA obtuvo mejores rutas (más cortas) que otras aplicaciones similares.

El modelo propuesto para el PRP con restricciones difusas permite a los usuarios considerar la imprecisión en las especificaciones de sus preferencias. Para resolver el problema, se propuso un enfoque basado en α-cortes. Esto permite al usuario lidiar con diferentes grados de satisfacción de las restricciones, pasando desde restricciones "crisp" a restricciones con una tolerancia máxima. Cuanta más tolerancia se permita, más relajado es el problema y se podría encontrar, potencialmente, una mejor ruta en términos de coste.

La idea de definir α-cortes es un método generalizado que consiste en elegir valores fijos de α al comienzo del proceso. Se ha propuesto un enfoque diferente más adaptativo en este trabajo. Consiste en elegir el grado de satisfacción de las restricciones de

forma iterativa, en lugar de fijar cada α-corte al comienzo del proceso, cada α-corte se elige de acuerdo con las soluciones ya conocidas. Se mostró la capacidad de este enfoque adaptativo para reducir la cantidad de soluciones duplicadas y para encontrar soluciones diferentes y potencialmente más variadas al resolver el mismo número de α-cortes.

El capítulo 4 se centra en el segundo sub-objetivo: los mapas adaptativos. Se definieron los mapas adaptativos y se propuso un modelo. La solución, un mapa adaptativo, consiste en la imagen de un mapa que se ha modificado para adaptarse a las preferencias del usuario. También se detalla un método para generar dicha imagen modificada. Como ejemplos, se han mostrado múltiples mapas adaptativos utilizando datos reales.

El sub-objetivo tres se abordó en el Capítulo 5, donde se propone un enfoque *a posteriori* para ayudar al tomador de decisiones (DM) a obtener un conjunto reducido de soluciones de interés (SOI) partiendo de un gran Conjunto de soluciones. En este trabajo, el concepto de "interés" dependía de las preferencias del DM.

Un parámetro, $\lambda \in [0, 1]$, indicaba un grado de filtrado en la generación del conjunto SOI. Este parámetro permite obtener conjuntos más pequeños cuando λ es mayor. Usando múltiples ejemplos, se mostró qué partes del frente de Pareto se mantuvieron bajo diferentes preferencias del DM y grados de filtrado (valores de λ).

El método propuesto fue capaz de lidiar con problemas de optimización tanto multiobjetivo como con muchos objetivos, ya que filtra las SOI del frente de Pareto independientemente del número de objetivos. Se pudo observar que, en general, la reducción de tamaño para el conjunto de SOI, con respecto al frente de Pareto original, fue significativa incluso con valores bajos de λ.

Un ejemplo de aplicación en Biología Sintética mostró que el método es versátil y podría ser una herramienta muy útil con múltiples campos de aplicación. En ese ejemplo específico, se demostró que era capaz de proporcionar las mejores rutas competitivas de acuerdo con las preferencias del DM y, en general, los conjuntos SOI proporcionados eran más fáciles de analizar que los conjuntos completos de soluciones iniciales. El enfoque parece prometedor según los expertos de los laboratorios de la Universidad de Newcastle que mostraron interés en las soluciones obtenidas sugiriendo realizar adicionalmente un análisis en laboratiorio de los resultados.

7.1 Trabajo futuro

A continuación se detallan mejoras significativas y posibles extensiones para la investigación desarrollada en este trabajo.

Problema de ruta personalizada

La definición de la escena peatonal se puede mejorar. Podría ser interesante explorar posibles métricas para evaluar cómo de verde es una ruta y otros atributos cualitativos como pueden ser "seguridad" o "entretenimiento". Los números difusos y las expresiones lingüísticas podrían ser útiles al modelar y gestionar esos atributos. A pesar de que su aplicación no se ha estudiado y explorado más a fondo en este trabajo, también podrían ser factibles dada la flexibilidad del modelo PRP presentado.

Una segunda mejora en esta cuestión es la consideración de atributos dinámicos y escenas. Por ejemplo, pensando en "seguridad", después de algunos años o incluso a diferentes horas del mismo día, la "seguridad" en una calle puede no ser la misma. La misma idea también se puede aplicar a atributos más comunes, como el tiempo de viaje, donde el tiempo necesario para completar una ruta puede variar de verano a invierno.

Además, es necesario que se describan más escenas, como rutas seguras o rutas en bicicleta. Esta descripción se puede lograr utilizando diferentes conjuntos de criterios que deben a su vez definirse de forma adecuada.

Finalmente, se puede explorar la combinación de restricciones difusas con variables lingüísticas difusas. Si las variables lingüísticas se ajustan al rango de valores posibles en la zona para los criterios, el usuario no necesita conocer las características del

mapa a priori para establecer una restricción factible, si no que puede elegir directamente una variable lingüística en su lugar.

Mapas adaptativos

A pesar de la calidad de las imágenes finales del Mapa Adaptativo, todavía hay margen de mejora en varias etapas. Como por ejemplo, en el cálculo de las nuevas ubicaciones de los puntos de interés. Actualmente, se usa el escalado multidimensional, pero se deben explorar otras técnicas como la de Geometría de distancia [60].

Otro aspecto es la visualización. Los métodos de triangulación y mapeo propuestos dan como resultado buenas imágenes de mapas en la práctica, pero sería interesante realizar una comparación adicional frente a otros métodos.

Se debe tener en cuenta que la matriz de disimilitud en el problema de escalado multidimensional debe ser simétrica. Eso significa que en caso de que la matriz subjetiva sea asimétrica, los resultados se obtienen teniendo en cuenta solo la mitad de los costes en la misma. El método se puede mejorar drásticamente, evitando este problema, resolviendo un problema de escalamiento multidimensional asimétrico [107].

Identificación de soluciones de interés

La propuesta actual sobre identificación de soluciones de interés (SOI) parte de un amplio conjunto de soluciones. Podría ser

interesante utilizar un conjunto de SOI obtenidos a partir de un método basado en geometría (como el método de la región de "rodilla") como punto de partida.

Podría ser interesante aplicar métodos de toma de decisiones multicriterio, una vez obtenido el conjunto de SOI, para clasificar las soluciones utilizando un modelo más sofisticado de las preferencias del DM (como la comparación entre pares de soluciones).

El uso de diferentes funciones de agregación para el cálculo de los intervalos supondría una mejora importante. Esta variación supondría la generación de un espacio de puntajes potenciales diferente.

Finalmente, sería útil implementar una herramienta de visualización con el objetivo de mostrar la relación entre las soluciones en el espacio de puntajes potenciales con el conjunto inicial de SOI. Dicha herramienta puede ayudar a comprender mejor la relación entre estos espacios.

References

[1] Ahn, B. S. and Park, K. S. (2008). Comparing methods for multiattribute decision making with ordinal weights. *Computers & Operations Research*, 35(5):1660–1670.

[2] Avelar, S. and Müller, M. (2000). Generating topologically correct schematic maps. In *In Proc. 9th Int. Symp. on Spatial Data Handling*, pages 4–28.

[3] Balstrøm, T. (2002). On identifying the most time-saving walking route in a trackless mountainous terrain. *Geografisk Tidsskrift-Danish Journal of Geography*, 102(1):51–58.

[4] Barron, F. H. and Barrett, B. E. (1996). Decision quality using ranked attribute weights. *Management Science*, 42(11):1515–1523.

[5] Beliakov, G., Pradera, A., and Calvo, T. (2007). *Aggregation Functions: A Guide for Practitioners*, volume 221. Springer Berlin Heidelberg.

[6] Bhattacharjee, K. S., Singh, H. K., and Ray, T. (2016). A study on performance metrics to identify solutions of interest from a trade-off set. In Ray, T., Sarker, R., and Li, X., editors, *Artificial Life and Computational Intelligence*, pages 66–77, Cham. Springer International Publishing.

[7] Bhattacharjee, K. S., Singh, H. K., Ryan, M., and Ray, T. (2017). Bridging the gap: Many-objective optimization and informed decision-making. *IEEE Transactions on Evolutionary Computation*, 21(5):813–820.

[8] Blečić, I., Cecchini, A., Congiu, T., Fancello, G., and Trunfio, G. A. (2014). Walkability explorer: An evaluation and design support tool for walkability. In *Computational Science and Its Applications – ICCSA 2014*, pages 511–521. Springer International Publishing.

[9] Borràs, J., Moreno, A., and Valls, A. (2014). Intelligent tourism recommender systems: A survey. *Expert Systems with Applications*, 41(16):7370–7389.

[10] Borst, H. C., de Vries, S. I., Graham, J. M., van Dongen, J. E., Bakker, I., and Miedema, H. M. (2009). Influence of environmental street characteristics on walking route choice of elderly people. *Journal of Environmental Psychology*, 29(4):477–484.

[11] Branke, J. (2016). MCDA and multiobjective evolutionary algorithms. In Greco, S., Ehrgott, M., and Figueira, J. R., editors, *Multiple Criteria Decision Analysis: State of the Art Surveys*, pages 977–1008. Springer New York.

[12] Branke, J., Deb, K., Dierolf, H., and Osswald, M. (2004). Finding knees in multi-objective optimization. In Yao, X., Burke, E. K., Lozano, J. A., Smith, J., Merelo-Guervós, J. J., Bullinaria, J. A., Rowe, J. E., Tiňo, P., Kabán, A., and Schwefel, H.-P., editors, *Parallel Problem Solving from Nature - PPSN VIII*, volume 3242, pages 722–731, Berlin, Heidelberg. Springer Berlin Heidelberg.

[13] Chen, Y., Yan, W., Li, C., Huang, Y., and Yang, L. (2018). Personalized optimal bicycle trip planning based on q-learning

algorithm. In *2018 IEEE Wireless Communications and Networking Conference (WCNC)*, pages 1–6. IEEE.

[14] Chuang, T.-N. and Kung, J.-Y. (2005). The fuzzy shortest path length and the corresponding shortest path in a network. *Computers & Operations Research*, 32(6):1409 – 1428.

[15] Coello, C. A. C., Lamont, G. B., and Van Veldhuizen, D. A. (2007). *Evolutionary Algorithms for Solving Multi-Objective Problems*, volume 5 of *Genetic and Evolutionary Computation Series*. Springer US, Boston, MA.

[16] Cox, T. F. and Cox, M. A. (2000). *Multidimensional scaling*. Chapman and hall/CRC.

[17] Craig, C. L., Brownson, R. C., Cragg, S. E., and Dunn, A. L. (2002). Exploring the effect of the environment on physical activity: A study examining walking to work. *American Journal of Preventive Medicine*, 23(2, Supplement 1):36 – 43.

[18] Cruz-Reyes, L., Fernandez, E., Sanchez, P., Coello, C. A. C., and Gomez, C. (2017). Incorporation of implicit decision-maker preferences in multi-objective evolutionary optimization using a multi-criteria classification method. *Applied Soft Computing*, 50:48 – 57.

[19] Da, Q.-L. and Liu, X.-W. (1999). Interval number linear programming and its satisfactory solution. *Systems Engineering Theory & Practice*, 19:3–7.

[20] Das, I. (1999). On characterizing the "knee" of the pareto curve based on normal-boundary intersection. *Structural optimization*, 18(2-3):107–115.

[21] Dey, A., Pradhan, R., Pal, A., and Pal, T. (2018). A genetic algorithm for solving fuzzy shortest path problems with interval type-2 fuzzy arc lengths. *Malaysian Journal of Computer Science*, 31(4):255–270.

[22] Dezfouli, M. B., Nadimi Shahraki, M. H., and Zamani, H. (2018). A novel tour planning model using big data. In *2018 International Conference on Artificial Intelligence and Data Processing (IDAP)*, pages 1–6. IEEE.

[23] Dou, Y., Zhu, L., and Wang, H. S. (2012). Solving the fuzzy shortest path problem using multi-criteria decision method based on vague similarity measure. *Applied Soft Computing*, 12(6):1621 – 1631.

[24] Dubois, D. and Prade, H. (1980). *Fuzzy Sets and Systems: Theory and Applications*, volume 144. Academic Press, New York.

[25] Enayattabar, M., Ebrahimnejad, A., and Motameni, H. (2018). Dijkstra algorithm for shortest path problem under interval-valued pythagorean fuzzy environment. *Complex & Intelligent Systems*, 5(2):93–100.

[26] Facchinetti, G., Ricci, R. G., and Muzzioli, S. (1998). Note on ranking fuzzy triangular numbers. *International Journal of Intelligent Systems*, 13(7):613–622.

[27] Feist, A. M., Henry, C. S., Reed, J. L., Krummenacker, M., Joyce, A. R., Karp, P. D., Broadbelt, L. J., Hatzimanikatis, V., and Palsson, B. Ø. (2007). A genome-scale metabolic reconstruction for escherichia coli k-12 mg1655 that accounts for 1260 orfs and thermodynamic information. *Molecular Systems Biology*, 3(1).

[28] Gastner, M. T., Seguy, V., and More, P. (2018). Fast flow-based algorithm for creating density-equalizing map projections. *Proceedings of the National Academy of Sciences*, 115(10):E2156–E2164.

[29] GraphHopper (2018). www.graphhopper.com. www.graphhopper.com. Accessed: 2019-09-06.

[30] Guo, Z. (2011). Mind the map! the impact of transit maps on path choice in public transit. *Transportation Research Part A: Policy and Practice*, 45(7):625–639.

[31] Gupta, H. and Wenger, R. (1997). Constructing piecewise linear homeomorphisms of simple polygons. *Journal of Algorithms*, 22(1):142 – 157.

[32] Guzmán, V. C., Pelta, D. A., and Verdegay, J. L. (2016). Fuzzy maximal covering location models for fighting dengue. In *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1–7. IEEE.

[33] Haklay, M. (2010). How good is volunteered geographical information? a comparative study of OpenStreetMap and ordnance survey datasets. *Environment and Planning B: Planning and Design*, 37(4):682–703.

[34] Henry, C. S., Zinner, J. F., Cohoon, M. P., and Stevens, R. L. (2009). iBsu1103: a new genome-scale metabolic model of bacillus subtilis based on SEED annotations. *Genome Biology*, 10(6):R69.

[35] Hernandes, F., Lamata, M. T., Verdegay, J. L., and Yamakami, A. (2007). The shortest path problem on networks with fuzzy parameters. *Fuzzy Sets and Systems*, 158(14):1561–1570.

[36] Hochmair, H. H. (2008). Grouping of optimized pedestrian routes for multi-modal route planning: A comparison of two cities. In *Lecture Notes in Geoinformation and Cartography*, pages 339–358. Springer Berlin Heidelberg.

[37] Hughes, E. (2007). Many-objective radar design software. http://code.evanhughes.org. Accessed: 2019-09-06.

[38] Ji, X., Iwamura, K., and Shao, Z. (2007). New models for shortest path problem with fuzzy arc lengths. *Applied Mathematical Modelling*, 31(2):259–269.

[39] Jiang, S., Torres, M., Pelta, D., Krabben, P., Daniel, R., Luzardo, J. T., Kaiser, M., and Krasnogor, N. (2018). Improving microbial strain design via multiobjective optimisation and decision making. In *AI for synthetic biology 2, IJCIA*, pages 1–6, Stockholm, Sweden.

[40] Jones, E., Oliphant, T., Peterson, P., et al. (2001–). SciPy: Open source scientific tools for Python. http://www.scipy.org/. Accessed: 2019-09-06.

[41] Karmakar, S. and Bhunia, A. K. (2012). A comparative study of different order relations of intervals. *Reliable Computing*, 16(1):38–72.

[42] Kasemsuppakorn, P. and Karimi, H. A. (2009). Personalised routing for wheelchair navigation. *Journal of Location Based Services*, 3(1):24–54.

[43] Kazemi, S. M., Rabbani, M., Tavakkoli-Moghaddam, R., and Shahreza, F. A. (2017). Blood inventory-routing problem under uncertainty. *Journal of Intelligent & Fuzzy Systems*, 32(1):467–481.

[44] Khademi, N. and Saedi, R. (2019). Latent learning and the formation of a spatiotemporal cognitive map of a road network. *Travel Behaviour and Society*, 14:66 – 80.

[45] Khan, J. A. and Alnuweiri, H. M. (2004). A fuzzy constraint-based routing algorithm for traffic engineering. In *IEEE Global Telecommunications Conference, 2004. GLOBECOM '04.*, volume 3, pages 1366–1372.

[46] Larichev, O. I. (1992). Cognitive validity in design of decision-aiding techniques. *Journal of Multi-Criteria Decision Analysis*, 1(3):127–138.

[47] Lee, D. T. and Schachter, B. J. (1980). Two algorithms for constructing a delaunay triangulation. *International Journal of Computer & Information Sciences*, 9(3):219–242.

[48] Leslie, E., Coffee, N., Frank, L., Owen, N., Bauman, A., and Hugo, G. (2007). Walkability of local communities: Using geographic information systems to objectively assess relevant environmental attributes. *Health & Place*, 13(1):111 – 122. Part Special Issue: Environmental Justice, Population Health, Critical Theory and GIS.

[49] Levin, V. I. (2004). Ordering of intervals and optimization problems with interval parameters. *Cybernetics and Systems Analysis*, 40(3):316–324.

[50] Li, D., Zeng, W., and Yin, Q. (2017). Novel ranking method of interval numbers based on the boolean matrix. *Soft Computing*, 22(12):4113–4122.

[51] Li, K., Chen, R., Min, G., and Yao, X. (2018). Integration of preferences in decomposition multiobjective optimization. *IEEE Transactions on Cybernetics*, 48(12):3359–3370.

[52] Liu, F., Pan, L.-H., Liu, Z.-L., and Peng, Y.-N. (2017). On possibility-degree formulae for ranking interval numbers. *Soft Computing*, 22(8):2557–2565.

[53] López-Jaimes, A. and Coello, C. A. C. (2014). Including preferences into a multiobjective evolutionary algorithm to deal with many-objective engineering optimization problems. *Information Sciences*, 277:1 – 20.

[54] López-Ornelas, E., Abascal-Mena, R., and Zepeda-Hernández, J. S. (2014). A geo-collaborative recommendation tool to help urban mobility. In *Human-Computer Interaction. Applications and Services*, pages 466–472. Springer International Publishing.

[55] Malhotra, N. K. (1982). Information load and consumer decision making. *Journal of Consumer Research*, 8(4):419–430.

[56] Manrique-Sancho, M.-T., Avelar, S., Iturrioz-Aguirre, T., and Manso-Callejo, M.-Á. (2018). Using the spatial knowledge of map users to personalize city maps: A case study with tourists in madrid,spain. *ISPRS International Journal of Geo-Information*, 7(8):332.

[57] Martins, E. Q. V. (1984). On a multicriteria shortest path problem. *European Journal of Operational Research*, 16(2):236–245.

[58] Moise, E. E. (1977). Piecewise linear homeomorphisms. In *Graduate Texts in Mathematics*, pages 42–45. Springer New York.

[59] Moore, R. E. and Bierbaum, F. (1979). *Methods and Applications of Interval Analysis*, volume 2 of *Studies in Applied*

and Numerical Mathematics. Soc for Industrial & Applied Math.

[60] Mucherino, A., Lavor, C., Liberti, L., and Maculan, N. (2012). *Distance geometry: theory, methods, and applications*. Springer Science & Business Media.

[61] Mukherjee, S. (2012). Dijkstra's algorithm for solving the shortest path problem on networks under intuitionistic fuzzy environment. *Journal of Mathematical Modelling and Algorithms*, 11(4):345–359.

[62] Naranjo, J. C. and Bayo, J. L. (2008). VADEO. http://www.vadeo.es/. Accessed: 2019-09-06.

[63] Okada, S. and Soper, T. (2000). A shortest path problem on a network with fuzzy arc lengths. *Fuzzy Sets and Systems*, 109(1):129–140.

[64] OpenStreetMap contributors (2017). Openstreetmap. http://www.openstreetmap.org/about. Accessed: 2019-09-06.

[65] Orth, J. D., Conrad, T. M., Na, J., Lerman, J. A., Nam, H., Feist, A. M., and Palsson, B. Ø. (2011). A comprehensive genome-scale reconstruction of, escherichia coli, metabolism—2011. *Molecular Systems Biology*, 7(1):535.

[66] Orth, J. D., Thiele, I., and Palsson, B. Ø. (2010). What is flux balance analysis? *Nature Biotechnology*, 28(3):245–248.

[67] Owen, N., Humpel, N., Leslie, E., Bauman, A., and Sallis, J. F. (2004). Understanding environmental influences on walking. *American Journal of Preventive Medicine*, 27(1):67–76.

[68] Pahlavani, P. and Delavar, M. R. (2014). Multi-criteria route planning based on a driver's preferences in multi-criteria

route selection. *Transportation Research Part C: Emerging Technologies*, 40:14–35.

[69] Pate, R. R. (1995). Physical activity and public health. *JAMA*, 273(5):402.

[70] Quercia, D., Schifanella, R., and Aiello, L. M. (2014). The Shortest Path to Happiness: Recommending Beautiful, Quiet, and Happy Routes in the City. In *Proceedings of the 25th ACM conference on Hypertext and social media*, pages 116—-125. ACM.

[71] R Core Team (2013). R: A language and environment for statistical computing. http://www.R-project.org/. Accessed: 2019-09-06.

[72] R team and contributors worldwide (2018). The R stats package. www.rdocumentation.org/packages/stats. Accessed: 2019-09-06.

[73] Raveau, S., Muñoz, J. C., and de Grange, L. (2011). A topological route choice model for metro. *Transportation Research Part A: Policy and Practice*, 45(2):138–147.

[74] Rolland-May, C. (1983). A valuation model of subjective spaces. *IFAC Proceedings Volumes*, 16(13):375 – 380. IFAC Symposium on Fuzzy Information, Knowledge Representation and Decision Analysis, Marseille, France, 19-21 July, 1983.

[75] Saelens, B. E., Sallis, J. F., and Frank, L. D. (2003). Environmental correlates of walking and cycling: Findings from the transportation,urban design,and planning literatures. *Annals of Behavioral Medicine*, 25(2):80–91.

[76] Sagar, B. S. D. (2013). Cartograms via mathematical morphology. *Information Visualization*, 13(1):42–58.

[77] Sengupta, A. and Pal, T. K. (2000). On comparing interval numbers. *European Journal of Operational Research*, 127(1):28 – 43.

[78] Soltani, A. and Fernando, T. (2004). A fuzzy based multiobjective path planning of construction sites. *Automation in Construction*, 13(6):717–734.

[79] Stillwell, W. G., Seaver, D. A., and Edwards, W. (1981). A comparison of weight approximation techniques in multiattribute utility decision making. *Organizational Behavior and Human Performance*, 28(1):62 – 77.

[80] Sun, H. and Yao, W. (2010). Comments on methods for ranking interval numbers. *Journal of systems engineering*, 3(005).

[81] Syarif, A., Muludi, K., Adrian, R., and Gen, M. (2018). Solving fuzzy shortest path problem by genetic algorithm. *IOP Conference Series: Materials Science and Engineering*, 332:012003.

[82] Tepper, N. and Shlomi, T. (2009). Predicting metabolic engineering knockout strategies for chemical production: accounting for competing pathways. *Bioinformatics*, 26(4):536–543.

[83] Torres, I., Cruz, C., and Verdegay, J. L. (2015). Solving the truck and trailer routing problem with fuzzy constraints. *International Journal of Computational Intelligence Systems*, 8(4):713–724.

[84] Torres, M. (2016). PRoA: A personalized route assistant application. https://play.google.com/store/apps/details?id=com.application.proa. Accessed: 2019-09-06.

[85] Torres, M., Jiang, S., Pelta, D., Kaiser, M., and Krasnogor, N. (2018a). Strain design as multiobjective network interdiction problem: A preliminary approach. In *Advances in Artificial Intelligence*, pages 273–282. Springer International Publishing.

[86] Torres, M., Pelta, D. A., Cruz, C., and Verdegay, J. L. (2017). Personalized route problem with fuzzy constraints. In *2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. IEEE.

[87] Torres, M., Pelta, D. A., and Lamata, M. T. (2018b). A new approach for solving personalized routing problems with fuzzy constraints. In *2018 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. IEEE.

[88] Torres, M., Pelta, D. A., and Verdegay, J. L. (2016). PRoA: una Aplicacion Android para el Diseño de Rutas Personalizadas. In *Congresos de la Asociación Española para la Inteligencia Artificial, Salamanca*.

[89] Torres, M., Pelta, D. A., and Verdegay, J. L. (2018c). PRoA: An intelligent multi-criteria personalized route assistant. *Engineering Applications of Artificial Intelligence*, 72:162–169.

[90] Torres, M., Pelta, D. A., and Verdegay, J. L. (2018d). A proposal for adaptive maps. In *Communications in Computer and Information Science*, pages 657–666. Springer International Publishing.

[91] Torres, M., Pelta, D. A., Verdegay, J. L., and Cruz, C. (2018e). Towards adaptive maps. *International Journal of Intelligent Systems*, 34(3):400–414.

[92] Tranter, M., Robineau, D., and Goodman, G. (2015). National travel survey: 2014. Technical report, Department for Transport, UK.

[93] Vansteenwegen, P. and Van Oudheusden, D. (2007). The mobile tourist guide: an OR opportunity. *OR insight*, 20(3):21–27.

[94] Verdegay, J. L. (1982). Fuzzy mathematical programming. In *Fuzzy information and decision processes*, volume 231, pages 231–237. North-Holland, Amsterdam.

[95] Walkonomics (2016). Walkonomics navigation & maps. https://play.google.com/store/apps/details?id=com.tasol.walkonomics. Accessed: 2019-09-06.

[96] Wang, H., Olhofer, M., and Jin, Y. (2017). A mini-review on preference modeling and articulation in multi-objective optimization: current status and challenges. *Complex & Intelligent Systems*, 3(4):233–245.

[97] Wang, Y.-M., Yang, J.-B., and Xu, D.-L. (2005). A two-stage logarithmic goal programming method for generating weights from interval comparison matrices. *Fuzzy sets and systems*, 152(3):475–498.

[98] Wu, W., Rui, Y., Su, F., Cheng, L., and Wang, J. (2014). Novel parallel algorithm for constructing delaunay triangulation based on a twofold-divide-and-conquer scheme. *GIScience & Remote Sensing*, 51(5):537–554.

[99] Xu, Y. and Da, Q. (2008). A method for multiple attribute decision making with incomplete weight information under uncertain linguistic environment. *Knowledge-Based Systems*, 21(8):837 – 841.

[100] Xu, Z. (2007). A method for multiple attribute decision making with incomplete weight information in linguistic setting. *Knowledge-Based Systems*, 20(8):719 – 725.

[101] Xu, Z. and Da, Q. L. (2003). Possibility degree method for ranking interval numbers and its application. *Journal of Systems Engineering*, 18:67–70.

[102] Xu, Z. S. and Da, Q. L. (2002). The uncertain OWA operator. *International Journal of Intelligent Systems*, 17(6):569–575.

[103] Zhang, X., Tian, Y., and Jin, Y. (2015). A knee point-driven evolutionary algorithm for many-objective optimization. *IEEE Transactions on Evolutionary Computation*, 19(6):761–776.

[104] Zhang, Y., Zhang, Z., Deng, Y., and Mahadevan, S. (2013). A biologically inspired solution for fuzzy shortest path problems. *Applied Soft Computing*, 13(5):2356–2363.

[105] Zheng, W., Liao, Z., and Qin, J. (2017). Using a four-step heuristic algorithm to design personalized day tour route within a tourist attraction. *Tourism Management*, 62:335–349.

[106] Zhou, L.-G., Chen, H.-Y., Merigó, J. M., and Gil-Lafuente, A. M. (2012). Uncertain generalized aggregation operators. *Expert Systems with Applications*, 39(1):1105–1117.

[107] Zielman, B. and Heiser, W. J. (1996). Models for asymmetric proximities. *British Journal of Mathematical and Statistical Psychology*, 49(1):127–146.

[108] Zielstra, D. and Hochmair, H. H. (2012). Using free and proprietary data to compare shortest-path lengths for effective pedestrian routing in street networks. *Transportation Research Record: Journal of the Transportation Research Board*, 2299(1):41–47.