Orienteering Problem: A Survey of Recent Variants, Solution Approaches and Applications

Article *in* European Journal of Operational Research · May 2016

DOI: 10.1016/j.ejor.2016.04.059

3 authors:

Aldy Gunawan
Singapore Management University

Hoong Chuin Lau
Singapore Management University

Pieter Vansteenwegen
KU Leuven

European Journal of Operational Research 255 (2016) 315–332

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Invited Review

Orienteering Problem: A survey of recent variants, solution approaches and applications

Aldy Gunawan[a,*], Hoong Chuin Lau[a], Pieter Vansteenwegen[b]

[a] School of Information Systems, Singapore Management University, 178902, Singapore
[b] KU Leuven, Leuven Mobility Research Center, CIB, Celestijnenlaan 300, Leuven 3001, Belgium

ARTICLE INFO

Article history:
Received 23 November 2015
Accepted 27 April 2016
Available online 6 May 2016

Keywords:
Scheduling
Survey
Orienteering Problem
Practical Applications

ABSTRACT

The Orienteering Problem (OP) has received a lot of attention in the past few decades. The OP is a routing problem in which the goal is to determine a subset of nodes to visit, and in which order, so that the total collected score is maximized and a given time budget is not exceeded. A number of typical variants has been studied, such as the Team OP, the (Team) OP with Time Windows and the Time Dependent OP. Recently, a number of new variants of the OP was introduced, such as the Stochastic OP, the Generalized OP, the Arc OP, the Multi-agent OP, the Clustered OP and others. This paper focuses on a comprehensive and thorough survey of recent variants of the OP, including the proposed solution approaches. Moreover, the OP has been used as a model in many different practical applications. The most recent applications of the OP, such as the Tourist Trip Design Problem and the mobile-crowdsourcing problem are discussed. Finally, we also present some promising topics for future research.

1. Introduction

The term “Orienteering Problem (OP)” was first introduced by Golden, Levy, and Vohra (1987). It is a combination of node selection and determining the shortest path between the selected nodes. The objective is to maximize the total score collected from visited (selected) nodes. In this problem, not all available nodes can be visited due to the limited time budget. Therefore, the OP can be seen as a combination between two classical combinatorial problems, the Knapsack Problem and the Travelling Salesman Problem (TSP) (Vansteenwegen, Souffriau, & Van Oudheusden, 2011a). Since then, several variants of the OP have been introduced, such as the Team OP (TOP), the (Team) OP with Time Windows ((T)OPTW) and the Time Dependent OP (TDOP).

Earlier surveys that situate the OP between other types of routing problems can be found in works of Feillet, Dejax, and Gendreau (2005) and Laporte and Rodríguez-Martín (2007). More recently, Vansteenwegen et al. (2011a) present a comprehensive survey about the OP and its variants, including problem descriptions, benchmark instances and solutions approaches. They also summarize some applications of the OP, such as the home fuel deliver problem and Tourist Trip Design Problem (TTDP). The survey covers research works up to the year 2009. Some possible future research lines have also been mentioned. Gavalas, Konstantopoulos, Mastakas, and Pantziou (2014a) summarize the use of the OP and its extensions to model single tour and multiple tour variants of the TTDP. Possible extensions of the OP that take into account more realistic TTDP issues or constraints are also described. Archetti, Speranza, and Vigo (2014c) present a survey of the broad class of vehicle routing problems with profits and consider the OP as the basic problem of this class. They also briefly cover other variants of the OP, such as the TOP and the Generalized OP. Archetti and Speranza (2014) provide a short survey of the Arc OP and the Team Orienteering Arc Routing Problem as the most recent arc routing problems with profits.

Given these previous surveys, the main contributions of this work are as follows:

- We extend the summary of the survey paper by Vansteenwegen et al. (2011a) by focusing on the most recent papers, not included in the previous surveys, about the OP and its variants including (T)OP, (T)OPTW and TDOP. This summary is presented in Section 2.
- We also extend the recent surveys by Archetti and Speranza (2014); Archetti et al. (2014c) by including 71 additional references to OP-related papers published since 2010 not found in both surveys. They are mainly discussed in Sections 2 and 3.
- We cover a number of new variants of the OP that have been published in the last five years, including the proposed solution

* Corresponding author. Tel.: +65 8158 6061.
E-mail addresses: aldygunawan@smu.edu.sg, aldy.gunawan2016@gmail.com (A. Gunawan), hclau@smu.edu.sg (H.C. Lau), pieter.vansteenwegen@kuleuven.be (P. Vansteenwegen).

http://dx.doi.org/10.1016/j.ejor.2016.04.059

approaches. For instance, the Generalized OP, the Stochastic OP, the Arc OP, the Multi-agent OP and others. They are discussed in Section 3.

- In Section 4, we extend the survey of the TTDP (Gavalas et al., 2014a) by including the most recent papers related to the application of the OP to the TTDP. We also present a number of recent applications and practical extensions of the OP, such as the mobile-crowdsourcing problem.
- Throughout the survey we provide additional insights and analyze the trends in the different variants of the OP by structuring and classifying the different variants, solution algorithms and applications. We also provide an overview of the available benchmark instances and discuss for which problems instances are not available. The resulting classification enables future researchers to find relevant literature and to analyze which characteristics and OP variants are most popular.
- Finally, we summarize promising topics for future research in Section 5.

2. Classical Orienteering Problem

In this section, a chronological summary of the most recent works related to the classical (T)OP and its variants, including (T)OPTW and TDOP, is presented. The focus lies on the most recent papers not mentioned in previous surveys. All benchmark instances for these variants including their characteristics are available via http://www.mech.kuleuven.be/en/cib/op.

Since we focus on the most recent contributions about the OP, we only present and briefly explain the basic mathematical model for the OP. For the other models, we refer the readers to the survey of Vansteenwegen et al. (2011a). Some mathematical models of more recent extensions of the OP will be presented and discussed in Section 3.

2.1. (Team) Orienteering Problem

The OP is defined as follows. Consider a set of nodes $N = \{1, \ldots, |N|\}$ where each node $i \in N$ is associated with the non-negative score S_i. The start and end nodes are fixed to nodes 1 and $|N|$, respectively. The goal of the OP is to determine a path, limited by a given time budget T_{max}, that visits a subset of N and maximizes the total collected score. It is assumed that collected scores can be added and that each node can be visited at most once. This is in contrast to other problems such as the Attractive Traveling Salesman Problem (Erdoğan, Cordeau, & Laporte, 2010) and other variants of the TSP (Punnen, 2007) where visiting a node close to a customer node is good enough to collect a portion of the score. The non-negative travel time between nodes i and j is represented as t_{ij}. The OP is extended to the TOP where the goal is to determine m paths, each limited by T_{max}, that maximize the total collected score.

The OP can be formulated as an integer programming model (Vansteenwegen et al., 2011a) with the following decision variables: $X_{ij} = 1$ if a visit to node i is followed by a visit to node j, and 0 otherwise. The variables u_i will be used in the subtour elimination constraints and allow to determine the position of the visited nodes in the path.

$$\text{Maximize} \sum_{i=2}^{|N|-1} \sum_{j=2}^{|N|} S_i X_{ij} \tag{1}$$

The objective function (1) is to maximize the total collected score.

$$\sum_{j=2}^{|N|} X_{1j} = \sum_{i=1}^{|N|-1} X_{i|N|} = 1 \tag{2}$$

Table 1
Benchmark OP and TOP instances.

Reference	Problem	Number of instances	Number of nodes $\|N\|$	Number of paths m
Tsiligirides (1984)	OP	18	32	1
		11	21	1
		20	33	1
Chao et al. (1996a)	OP	26	66	1
		14	64	1
Fischetti et al. (1998)	OP	3 × 15	21 to 262	1
		3 × 44	47 to 400	1
		4 × 11	25 to 500	1
		5 × 15	21 to 301	1
Chao et al. (1996b)	TOP	3 × 18	32	2 to 4
		3 × 11	21	2 to 4
		3 × 20	33	2 to 4
		3 × 20	100	2 to 4
		3 × 26	66	2 to 4
		3 × 14	64	2 to 4
		3 × 20	102	2 to 4
Dang et al. (2013b)	TOP	333	100 - 399	2 to 4

Constraints (2) ensure that the path starts from node 1 and ends on $|N|$.

$$\sum_{i=1}^{|N|-1} X_{ik} = \sum_{j=2}^{|N|} X_{kj} \le 1; \ \forall k = 2, \ldots, (|N|-1) \tag{3}$$

Constraints (3) ensure the connectivity of the path and guarantee that each node is visited at most once.

$$\sum_{i=1}^{|N|-1} \sum_{j=2}^{|N|} t_{ij} X_{ij} \le T_{max} \tag{4}$$

Constraint (4) limits the total travel time within the time budget T_{max}.

$$2 \le u_i \le |N|; \ \forall i = 2, \ldots, |N| \tag{5}$$

$$u_i - u_j + 1 \le (|N|-1)(1 - X_{ij}); \ \forall i = 2, \ldots, |N| \tag{6}$$

The combination of constraints (5) and (6) prevents subtours (Miller, Tucker, & Zemlin, 1960).

2.1.1. Benchmark instances

Vansteenwegen, Souffriau, Vanden Berghe, and Van Oudheusden (2011b) summarize three groups of benchmark OP instances which are available from Tsiligirides (1984), Chao, Golden, and Wasil (1996a) and Fischetti, Salazar-González, and Toth (1998). For the TOP, benchmark instances are available from Chao, Golden, and Wasil (1996b). Dang, Guibadj, and Moukrim (2013b) introduce a new set of larger instances for the TOP. This set is based on the OP instances of Fischetti et al. (1998) with recalculating the time budget per vehicles. The time budget per path is calculated by dividing the time budget of the original OP instance with the number of vehicles of the new TOP instance. In total, there are 333 new instances which can be accessed via https://www.hds.utc.fr/~moukrim/dokuwiki/en/top.

Table 1 summarizes the available benchmark OP and TOP instances. For more details about the characteristics of these sets of benchmark instances, we refer to the survey of Vansteenwegen et al. (2011b) and the original papers.

2.1.2. Solution approaches

Table 2 outlines the most recent papers for both OP and TOP, including the proposed algorithms and benchmark instances. The

Table 2
Papers on OP and TOP published since 2010.

Reference	Problem	Algorithm	Benchmark instances	Performance
Sevkli and Sevilgen (2010b)	OP	Strengthened Particle Swarm Optimization	Tsiligirides (1984)	No Improvement
Sevkli and Sevilgen (2010a)	OP	Discrete Strengthened Particle Swarm Optimization	Tsiligirides (1984); Chao et al. (1996a)	Improve 1 best known solution
Chekuri et al. (2012)	OP	Approximation algorithms	-	-
Liang et al. (2013)	OP	Multi-Level Variable Neighborhood Search	Three new sets of larger size instances (not available online)	No Improvement
Campos et al. (2014)	OP	Greedy Randomized Adaptive Search Procedure and Path Relinking	Tsiligirides (1984); Chao et al. (1996a); Fischetti et al. (1998)	No Improvement
Marinakis et al. (2015)	OP	Memetic-Greedy Randomized Adaptive Search Procedure	Tsiligirides (1984); Chao et al. (1996a)	No improvement
Poggi et al. (2010)	TOP	Branch-cut-and-price algorithm	Chao et al. (1996b)	No Improvement
Bouly et al. (2010)	TOP	Memetic Algorithm	Chao et al. (1996b)	Improve 5 best known solutions
Muthuswamy and Lam (2011)	TOP	Discrete Particle Swarm Optimization	Chao et al. (1996b)	No Improvement
Dang et al. (2011)	TOP	Particle Swarm Optimization-based Memetic Algorithm	Chao et al. (1996b)	No Improvement
Dang et al. (2013a)	TOP	Branch-and-cut algorithm	Chao et al. (1996b)	Improve 29 best known solutions
Dang et al. (2013b)	TOP	Particle Swarm Optimization-inspired Algorithm	Chao et al. (1996b); new sets of larger instances	Improve 1 best known solution
Lin (2013)	TOP	Multi-start Simulated Annealing	Chao et al. (1996b)	Improve 5 best known solutions
Ferreira et al. (2014)	TOP	Genetic Algorithm	Chao et al. (1996b)	No Improvement
Keshtkaran et al. (2016)	TOP	Branch-and-price algorithm and Branch-and-cut-and-price algorithm	Chao et al. (1996b)	Improve 17 best known solutions
Ke et al. (2015)	TOP	Pareto mimic algorithm	Chao et al. (1996b); Dang et al. (2013b)	Improve 10 best known solutions

performance of the algorithms is described in detail in the text and summarized in Table 2.

Sevkli and Sevilgen (2010b) introduce a Strengthened Particle Swarm Optimization (StPSO) algorithm. The main modification is on pioneering-particles which achieves the swarm's experience. Each pioneer-particle is processed in two steps, initiating an external local search and assigning a random velocity. By introducing two steps of modification, the exploration mechanism of PSO is further improved and premature convergence can be avoided. Two sub-variants by only including one of two possible steps, namely Diversification Strengthened PSO (DS-PSO) and Intensification Strengthened PSO (IS-PSO), are also introduced.

Sevkli and Sevilgen (2010a) propose a discrete PSO algorithm for the OP, namely Discrete Strengthened PSO (DStPSO). The main idea is to focus on modifying the pioneering-particle that achieves or enhances the best solution by employing Reduced Variable Neighborhood Search (RVNS). Other particles continue to search the solution space as in the standard PSO algorithm.

Chekuri, Korula, and Pál (2012) study the OP in the context of undirected and directed graphs. A (2+δ)-approximation algorithm is proposed for undirected graphs, where $\delta > 0$. For the OP in directed graphs, it is proven that there is an $O(\log^2 \text{OPT})$ approximation algorithm, where OPT is the length of an optimum k-TSP tour.

Liang, Kulturel-Konak, and Lo (2013) develop another type of VNS, namely Multi-Level VNS (ML-VNS). In the ML-VNS algorithm, certain identical instances will be solved concurrently so computational and search resources can be reduced. Three sets of large-sized OP instances are used in order to compare the ML-VNS algorithm against Tabu Search (TS) (Kulturel-Konak, Norman, Coit, & Smith, 2004) and Probabilistic Solution Discovery Algorithm (PSDA) (Ramirez-Marquez, Kulturel-Konak, & Sanseverino, 2010). The first set of instances they used can be found in the work of Ramirez-Marquez et al. (2010) while the other two sets are newly generated instances with larger number of nodes which are not available online. PSDA is only applied to the new instance sets. ML-VNS outperforms both TS and PSDA in terms of the solution quality for all sets of instances. On the other hand, ML-VNS requires more computational time than TS does.

Another method to solve the OP is proposed by Campos, Martí, Sánchez-Oro, and Duarte (2014). The method is based on the Greedy Randomized Adaptive Search Procedure (GRASP) and the Path Relinking (PR) methodologies (Resende & Ribeiro, 2003). Four different constructive methods and two neighborhoods in the local search of GRASP are explored. PR is then adapted in the context of GRASP. The latest algorithm, namely Memetic-GRASP (MemGRASP), is proposed by Marinakis, Politis, Marinaki, and Matsatsinis (2015). The algorithm combines three algorithms: GRASP, evolutionary algorithm and two local search procedures (2-OPT and EXCHANGE). MemGRASP solves only 87 benchmark instances (Chao et al., 1996a; Tsiligirides, 1984).

The TOP was initially introduced by Chao et al. (1996b) by extending the OP to multiple paths. Poggi, Viana, and Uchoa (2010) propose three different mathematical formulations for the TOP. An exact algorithm, a robust branch-and-cut-and-price algorithm, is proposed for solving the TOP. The pricing sub-problem is solved by Dynamic Programming (DP). Two different cuts used in the proposed algorithm are the Min Cut inequalities and the Triangle Clique cuts, adopted from the work of Pessoa, Poggi, and Uchoa (2009).

Bouly, Dang, and Moukrim (2010) propose a Memetic Algorithm (MA) that combines GA and some local search techniques. The encoding process uses an Optimal Split procedure (Ulusoy, 1985). Local search techniques, such as SHIFT, SWAP and REPAIR operators, are used as mutation operators. The results obtained

Table 3
Benchmark OPTW and TOPTW instances.

Reference	Name	Instance Sets	Number of nodes $\|N\|$	Number of paths m
Righini and Salani (2009)	Solomon	c100, r100, rc100	100	1 to 4
	Cordeau	pr01 - pr10	48 to 288	
Montemanni and Gambardella (2009)	Solomon	c200, r200, rc200	100	1 to 4
	Cordeau	pr11 - pr20	48 to 288	
Vansteenwegen et al. (2009)	Solomon	c100, r100, rc100	100	3 to 20
		c200, r200, rc200	100	
	Cordeau	pr01 - pr10	48 to 288	
Gavalas et al. (2013)	t^*	t100	101	1 to 3
		t200	119	

by solving benchmark TOP instances are compared with state-of-the art algorithms, and is shown to be comparable with those methods.

A population based meta-heuristic, namely Discrete PSO (DPSO), is proposed by Muthuswamy and Lam (2011) for solving the TOP with the number of paths, m, from 2 to 4. The DPSO algorithm consists of two steps: initial solution construction and particle update procedure. The initial solution for (m-1) paths is randomly generated, while the one of the last path is generated based on a combination of an s/d (score/distance) technique and a random generation procedure. In the particle update procedure, the Reduced VNS (RVNS) and 2-Opt are introduced to improve the solution quality.

Another PSO-based MA (PSOMA) is proposed by Dang, Guibadj, and Moukrim (2011). The algorithm extends the MA (Bouly et al., 2010) by changing the global scheme to PSO. Dang et al. (2013b) extend the previous work of Dang et al. (2011) by proposing an effective PSO-inspired Algorithm (PSOiA) for the TOP. Giant tours to indirectly encode particle positions are used. The evaluation process is based on an interval graph model so more iterations can be done without increasing the global computational time. The proposed algorithm is able to examine larger neighborhoods and to explore the search space in a faster way.

A branch-and-cut algorithm is introduced by Dang, El-Hajj, and Moukrim (2013a). This exact algorithm is based on a linear formulation with a polynomial number of binary variables. A new set of dominance properties and valid inequalities is included. Lin (2013) proposes a Multi-start Simulated Annealing (MSA) by hybridizing advantages of SA and the multi-start hill climbing strategy. By including the multi-start hill climbing strategy, the possibility of getting trapped in a local optima is minimized. Ferreira, Quintas, and Oliveira (2014) introduce a GA approach, namely GATOP, that imitates the natural process of evolution to solve the TOP. Solutions are generated by using nature-inspired techniques such as mutation, crossover, inheritance and selection.

Keshtkaran, Ziarati, Bettinelli, and Vigo (2016) propose a branch-and-price algorithm to find proven optimal solutions for the TOP. The algorithm is based on the one proposed by Boussier, Feillet, and Gendreau (2007) with some additional novel features, such as the algorithm used for solving the pricing sub-problem, new relaxations of the pricing sub-problem and valid inequalities. In addition, they propose a branch-and-cut-and-price algorithm using subset-row inequalities.

The latest approach to solve the TOP, called Pareto mimic algorithm (PMA), is proposed by Ke, Zhai, Li, and Chan (2015). It uses a new operator, a mimic operator, to generate a new solution by imitating an incumbent solution. It also adopts a new operator swallow in order to swallow (or insert) an infeasible node and then repair the resulting infeasible solution.

2.2. (Team) Orienteering Problem with Time Windows

The OP with Time Windows (OPTW) considers the time window constraints that arise in the context when the service at a particular node has to start within a predefined time window (Labadie, Mansini, Melechovský, & Wolfler Calvo, 2012). Each node is assigned a time window $[O_i, C_i]$ and a visit to a node can only start during this time window. An early arrival to a particular node leads to waiting times, while a late arrival causes an infeasibility issue.

The OPTW assumes the number of paths m is equal to one. Kantor and Rosenwein (1992) started the initial investigation of the OPTW. The OPTW is extended to the TOPTW when $m > 1$.

2.2.1. Benchmark instances

The benchmark instances (Vansteenwegen, Souffriau, Vanden Berghe, & Van Oudheusden, 2009) are listed in Table 3. Vansteenwegen et al. (2009) introduce more difficult instances for the TOPTW, but nonetheless, the optimal solution is known for all these instances. The optimal solution for each instance is the sum of all scores.

Gavalas, Konstantopoulos, Mastakas, Pantziou, and Tasoulas (2013) introduce another set of benchmark TOPTW instances with different characteristics: (a) nodes are associated with much wider, overlapping and multiple time windows; (b) nodes are densely located at certain areas, while isolated nodes are few; (c) visiting time at a node is typically correlated with its profit value; and (d) the time budget is typically in order of a few hours per day. The instances are available at http://www2.aegean.gr/dgavalas/public/op_instances/.

2.2.2. Solution approaches

Again we start our chronological overview of solution approaches after the survey of Vansteenwegen et al. (2011a). Table 4 presents the most recent papers for both OPTW and TOPTW. The performances of the proposed algorithms in improving the best known solutions are also highlighted. All proposed algorithms are tested on the first three benchmark OPTW and TOPW instances (Table 3), except the algorithms proposed by Duque, Lozano, and Medaglia (2015) and Gunawan, Lau, and Lu (2015a) since both are only applied to the OPTW. Take note that only the Cluster Search Cluster Ratio (CCSCRatio) and Cluster Search Cluster Routes (CSCRoutes) algorithms (Gavalas et al., 2013) are applied to t^* instances.

Table 5 summarizes the performance of each algorithm in obtaining the best known solutions at the time the algorithm was published. We exclude the ones proposed by Duque et al. (2015) and Gunawan et al. (2015a). We also exclude the results by

Table 4
Papers on OPTW and TOPTW published since 2010.

Reference	Problem	Algorithm	Performance
Labadie et al. (2011)	OPTW and TOPTW	Hybrid Greedy Randomized Adaptive Search Procedure and Evolutionary Local Search	Improve 150 best known solutions
Gambardella et al. (2012)	OPTW and TOPTW	Enhanced Ant Colony System	Improve 26 best known solutions
Lin and Yu (2012)	OPTW and TOPTW	fast SA and slow SA	Improve 33 best known solutions
Labadie et al. (2012)	OPTW and TOPTW	LP-based Granular Variable Neighborhood Search	Improve 25 best known solutions
Souffriau et al. (2013)	OPTW and TOPTW	Hybrid Greedy Randomized Adaptive Search Procedure and Iterated Local Search	Improve 3 best known solutions
Gavalas et al. (2013)	OPTW and TOPTW	Cluster Search Cluster Ratio and Cluster Search Cluster Routes	No improvement
Hu and Lim (2014)	OPTW and TOPTW	Hybrid Local Search and Simulated Annealing	Improve 35 best known solutions
Cura (2014)	OPTW and TOPTW	Artificial Bee Colony	No improvement
Duque et al. (2015)	OPTW	Pulse algorithm	No improvement
Gunawan et al. (2015a)	OPTW	Iterated Local Search	Improve 8 best known solutions
Gunawan et al. (2015c)	OPTW and TOPTW	Well-Tuned Iterated Local Search	Improve 31 best known solutions
Gunawan et al. (2015b)	OPTW and TOPTW	Hybrid Simulated Annealing and Iterated Local Search	Improve 19 best known solutions

Table 5
Percentage of best known solutions obtained by the algorithm.

Reference	Algorithm	Number of instances per m	Percentage of best known solutions				Average
			$m = 1$	$m = 2$	$m = 3$	$m = 4$	
Labadie et al. (2011)	GRASP-ELS	76	50.0	21.1	32.9	46.1	37.5
Lin and Yu (2012)	SSA	76	51.3	34.2	39.5	56.6	45.4
Labadie et al. (2012)	GVNS	76	36.8	30.3	40.8	44.7	38.2
Souffriau et al. (2013)	GRILS	76	51.3	15.8	22.4	39.5	32.2
Hu and Lim (2014)	I3CH	76	43.4	34.2	57.9	55.3	47.7
Cura (2014)	ABC	76	48.7	36.8	46.1	48.7	45.1
Gunawan et al. (2015c)	ILS	76	68.4	51.3	56.6	55.3	57.9
Gunawan et al. (2015b)	SAILS	76	67.1	50.0	57.9	53.9	57.2

Gambardella, Montemanni, and Weyland (2012) and Gavalas et al. (2013) since the details are not available. We calculate the percentage of best known solutions obtained for each m by each algorithm. Most algorithms perform well for $m = 1$ and 4. It seems that $m = 2$ is the most difficult to solve. We observe that Iterated Local Search (ILS) (Gunawan, Lau, & Lu, 2015c) and Hybrid Simulated Annealing and ILS (SAILS) (Gunawan, Lau, & Lu, 2015b) are currently the state-of-the-art algorithms for the OPTW and TOPTW problems. Both algorithms are able to obtain more than 50 percent of the best known solutions (see Table 5).

A hybridization of a Greedy Randomized Adaptive Search Procedure (GRASP) and an Evolutionary Local Search (ELS) algorithm, namely GRASP-ELS, is proposed by Labadie, Mansini, Melechovskỳ, and Wolfler Calvo (2011). Five simple constructive heuristics are introduced in order to build the initial solutions. The first three heuristics, which are focusing on adding one node into the solution at each iteration, are different in the way of selecting the best insertion. The other two heuristics are based on sweep algorithms that focus on creating clusters of nodes and constructing a tour for each cluster. All different constructive heuristics are used in GRASP to generate distinct initial solutions that would be further improved by the ELS algorithm.

Gambardella et al. (2012) identify the drawbacks of the Ant Colony System (ACS) they designed earlier (Montemanni & Gambardella, 2009) and propose an Enhanced ACS (EACS) algorithm to overcome the drawbacks. Two operations to enhance the performance of ACS are: (1) considering the best solution found so far during the construction phase, and (2) applying the local search procedure only on those solutions on which the local search has not been recently applied. EACS is able to improve the results of ACS on average.

Two different Simulated Annealing algorithms, fast SA (FSA) and slow SA (SSA), are developed in order to tailor two different scenarios (Lin & Yu, 2012). The former, which is based on shorter computational times, is mainly for the applications that require quick responses. On the other hand, the latter is more concerned about the quality of the solutions; therefore, it will only be terminated if the current best solution has not been improved for a certain number of consecutive temperature values. Both implement a standard SA procedure with a random neighborhood structure that features SWAP, INSERTION and INVERSION operations.

Labadie et al. (2012) introduce an LP-based Granular Variable Neighborhood Search (GVNS) to solve both OPTW and TOPTW. The idea of granularity that includes time constraints and profits in addition to pure distances is introduced. The granularity aims at reducing the size of the analyzed neighborhoods without losing its effectiveness. The dual optimal solutions of an LP-problem are used to construct granular neighborhoods.

Souffriau, Vansteenwegen, Vanden Berghe, and Van Oudheusden (2013) introduce a hybrid algorithm based on GRASP and ILS, namely GRILS, for solving the Multi-Constraint Team Orienteering Problem with Multiple Time Windows (MC-TOPMTW), a variant of the TOPTW. In this problem, some nodes may have one or more time windows. A certain number of additional knapsack constraints are included in the problem. Since there are no benchmark instances for MC-TOPMTW, the performance of the proposed algorithm is evaluated by using three related problems, the TOPTW, the Selective Vehicle Routing Problem with Time Windows (SVRPTW) (Boussier et al., 2007) and the Multi-Constraint Team Orienteering Problem with Time Windows (MC-TOPTW) (Garcia, Vansteenwegen, Arbelaitz, Souffriau, & Linaza, 2013).

Gavalas et al. (2013) focus on the Tourist Trip Design Problem (TTDP) as the application of the TOPTW. They describe the limitations of ILS (Vansteenwegen et al., 2009) in solving the TTDP. Therefore, they introduce two cluster-based algorithms, namely Cluster Search Cluster Ratio (CSCRatio) and Cluster Search Cluster Routes (CSCRoutes). The main incentive is to motivate visits to topology areas featuring high density of good candidate nodes.

Both algorithms organize POIs into clusters based on topological distance criteria. Unlike CSCRatio, CSCRoutes is designed to construct paths that each cluster can only be visited at most once.

An iterative framework (I3CH) which is based on two components, a local search (LS) procedure and SA, is proposed by Hu and Lim (2014). Both components are used to explore the solution space and to discover a set of paths. The eliminator is mainly used to generate neighborhood solutions. Both LS and SA send the solutions into *POOL*. The last component of the framework, namely Route Recombination (RR), which focuses on combining the paths from *POOL* to identify high quality solutions is included. RR solves a set packing formulation to produce the best combination. The final solution is used to assist LS and SA. The entire framework is run iteratively within a certain number of iterations.

Cura (2014) proposes an Artificial Bee Colony (ABC) approach. A new food source acceptance criterion based on SA and a new scout bee search behavior based on a local search procedure are incorporated in order to improve the solution quality of benchmark TOPTW instances.

Duque et al. (2015) adapt the pulse algorithm (Lozano, Duque, & Medaglia, 2015) for solving the OPTW. The pulse algorithm is a general-purpose pulse framework for hard shortest path problems. The algorithm is tested on a subset of benchmark OPTW instances.

ILS is proposed to solve the OPTW by Gunawan et al. (2015a). The algorithm is started by generating an initial feasible solution using a greedy construction heuristic. A set of feasible candidate nodes to be inserted is generated. The insertion of unscheduled nodes is based on the Roulette-Wheel selection (Goldberg, 1989). The initial solution obtained is further improved by ILS. ILS is mainly based on several local search components, such as SWAP, 2-OPT, INSERT and REPLACE. A combination between ACCEPTANCECRITERION and PERTURBATION mechanisms is implemented to control the balance between diversification and intensification of the search.

Gunawan et al. (2015c) extend ILS by including more local search operations, such as SWAP and MOVE (Gunawan et al., 2015a). SWAP is done by exchanging two nodes within a particular path. This is extended by exchanging two nodes between two paths. In PERTURBATION, another step, namely EXCHANGEROUTE, is included. The idea of EXCHANGEROUTE is to swap all nodes from two different paths. A more general mathematical model for the TOPTW is also introduced. The model can accommodate different scenarios, such as different start and end nodes for a particular path, each path may have a different time budget. The ILS is also applied to generate paths for more general real-world problems where above-mentioned scenarios do exist.

A hybridization of SA and ILS, namely SAILS, is proposed by Gunawan et al. (2015b) to avoid the disadvantage of early termination in ILS. SA has the capability to escape from a local optimum by accepting a worse solution with a small probability.

2.3. Time Dependent Orienteering Problem

In the OP, the travel time between two nodes is assumed to be a constant value. However, in many practical situations, the travel time actually depends on the network properties, such as congestion levels, construction zones on certain links, and so on, which may affect the travel time between two nodes. The most common example is the travel time of public transport including the waiting time factor (Garcia et al., 2013). The OP where the travel time between two nodes depends on the departure time at the first node is called the Time Dependent OP (TDOP) (Verbeeck, Sörensen, Aghezzaf, & Vanssteenwegen, 2014a). Fomin and Lingas (2002) provide a formal definition of the TDOP and state it is NP-hard.

We present a mixed integer programming model for the TDOP (Verbeeck et al., 2014a). The model extends the OP model described in Section 2.1. The decision variables and parameters used are listed below. The details about calculating parameters θ_{ijt}, η_{ijt} and T_{ij} can be referred to the work of Verbeeck et al. (2014a).

- $X_{ijt} = 1$: if a vehicle travels from node i to j with a departure time in time slot t, 0 otherwise
- W_{ijt}: the departure time in time slot t when travelling from node i to j
- θ_{ijt}: slope coefficient of the linear time-dependent travel time
- η_{ijt}: intercept coefficient of the linear time-dependent travel time
- τ_{ijt}: lower limit of time slot t for arc (i, j)
- T_{ij}: number of time slots for arc (i, j)

$$\text{Maximize} \sum_{i=2}^{|N|-1} \sum_{j=2}^{|N|} \sum_{t=1}^{T_{ij}} S_i X_{ijt} \tag{7}$$

The objective function (7) is to maximize the total collected score.

$$\sum_{j=2}^{|N|} X_{1j1} = \sum_{i=1}^{|N|-1} \sum_{t=1}^{T_{i|N|}} X_{i|N|t} = 1 \tag{8}$$

Constraints (8) ensure that the path starts and ends at nodes 1 and $|N|$, respectively.

$$\sum_{i=1}^{|N|-1} \sum_{t=1}^{T_{ih}} X_{iht} = \sum_{j=2}^{|N|} \sum_{t=1}^{T_{hj}} X_{hjt} \leq 1; \ \forall h = 2, \ldots, |N| - 1 \tag{9}$$

Constraints (9) ensure each node is visited at most once.

$$\sum_{i=1}^{|N|-1} \sum_{t=1}^{T_{ih}} [W_{iht} + (\theta_{iht} W_{iht} + \eta_{iht} X_{iht})] = \sum_{j=2}^{|N|} \sum_{t=1}^{T_{hj}} W_{hjt}; \ \forall h = 2, \ldots, |N| - 1 \tag{10}$$

Constraints (10) guarantee that the departure time of a succeeding node is equal to the sum of the departure time of the previous node together with the travel time between these two nodes. These constraints do not allow waiting time.

$$X_{ijt} \tau_{ijt} \leq W_{ijt} \leq X_{ijt} \tau_{ij(t+1)}; \ i = 1, \ldots, |N| - 1, j = 2, \ldots, |N| \ \forall t \tag{11}$$

$$\sum_{i=1}^{|N|-1} \sum_{j=2}^{|N|} \sum_{t=1}^{T_{ij}} \left[\theta_{ijt} W_{ijt} + \eta_{ijt} X_{ijt} \right] \leq T_{max} \tag{12}$$

Constraints (11) categorize the departure time in the right time slot which is done by multiplying the departure time with its corresponding θ and η in constraint (12). Constraint (12) also enforces the limited travel time.

$$W_{1i1} = 0; \ \forall i = 1, \ldots, |N| \tag{13}$$

Constraints (13) ensure a path starts in time slot one.

$$0 \leq W_{ijt} \leq T_{max}; \ \forall t, i, j = 1, \ldots, |N| \tag{14}$$

Constraints (14) guarantee that all departure times are less than or equal to T_{max}.

2.3.1. Benchmark instances

Verbeeck et al. (2014a) create data sets for the TDOP based on the TOP instances (Chao et al., 1996b; Vansteenwegen et al., 2011a). Inputs required for this data are the arc category matrix, the speed matrix and the adapted TOP instances. The arc category matrix defines the arc category of every arc connecting two nodes based on one of the five categories. The speed matrix contains a speed level for each time slot and related arc category.

Table 6
Benchmark TDOP instances.

Reference		Number of instances	Number of nodes $\|N\|$
		9	32
		9	21
		9	33
Verbeeck et al. (2014a)		10	100
		3	66
		9	64
		10	102
Gunawan et al. (2014)	MODIFIED	7 × 4	21 to 102
	RANDOM	4 × 4	10 to 40
	REAL	2 × 1	17 to 40

Gunawan, Yuan, and Lau (2014) introduce modified benchmark instances adopted from Verbeeck et al. (2014a) (MODIFIED), randomly generated instances (RANDOM) and two real-world instances (REAL). For the modified benchmark instances, the original travel time is discretized by different intervals (in minutes). The randomly generated instances are generated by varying values of $|N|$ and T_{max}. Two real-world instances are obtained from two popular theme parks in Asia. Each node represents an attraction or ride, the score of each attraction is derived from user preference data. Other data such as the service and waiting times are assumed to be known.

Table 6 presents an overview of the available benchmark TDOP instances. All instances are available in http://www.mech.kuleuven.be/en/cib/op#section-23 and http://centres.smu.edu.sg/larc/Orienteering-Problem-Library/. While many different works have dealt with benchmark (T)OP and (T)OPTW instances, it is worth mentioning that benchmark TDOP instances are not widely used. Until now, each paper focuses on its own instances.

2.3.2. Solution approaches

Table 7 summarizes the algorithms proposed for solving the TDOP and its variants, including the applications. Li (2012) studies the TDOP in the context of a transportation network with given source and destination nodes. The proposed algorithm is based on the idea of network planning and dynamic node labeling programming.

Verbeeck et al. (2014a) propose an algorithm based on the combination of an Ant Colony System (ACS) with a time-dependent local search procedure equipped with a local evaluation metric. Both intensification and diversification steps are implemented. The strength of the insertion step is in the fast evaluation of the possible insertion of a node. The diversification mechanism is done by depreciating the pheromone trails during the construction procedure. A tool to solve any TDOP instance can be downloaded at http://www.mech.kuleuven.be/en/cib/op#section-23.

Gunawan et al. (2014) study the TDOP based on the real-life application of providing automatic tour guidance to a large leisure facility. The problem is formulated and solved as an ILP model. To obtain solutions within acceptable computational time, four different metaheuristics are developed and compared: a restart greedy append construction heuristic (Greedy), a restart Variable Neighborhood Descent (VND), a basic version of ILS (Basic ILS) and a modified ILS with adaptive perturbation size and probabilistically intensified restart (Adaptive ILS).

The Time Dependent OP with Time Windows (TDOPTW) is another extension of the TDOP by adding time windows to each node. One application of the TDOPTW is the Personalized Electronic Tourist Guides (PET) (Garcia, Arbelaitz, Vansteenwegen, Souffriau, & Linaza, 2010; Garcia et al., 2013). The PET focuses on maximizing the tourists' satisfaction in near real-time by taking several restrictions into consideration, such as opening hours, duration of the visits and travel time. It is assumed that the travel time between nodes depends on the leave time of the first node and the transportation mode.

Garcia et al. (2010) integrate the public transportation that influences the travel time between two nodes. A hybrid algorithm combining two heuristics is introduced to solve the problem. The first heuristic focuses on the calculation of the average travel time between all pairs of nodes. Based on these averages a solution is calculated and, at the end, a repair procedure is introduced to incorporate the real travel times between two nodes. The second heuristic implements ILS (Vansteenwegen et al., 2009).

Abbaspour and Samadzadegan (2011) formulate the TTDP in a large urban area in the context of the TDOPTW. The problem includes three different modes of transportation that effect the travel time between POIs: walking, bus and subway. In order to solve the problem, an evolutionary strategy based on the Genetic Algorithm (GA) is proposed. Chromosomes with variable lengths are used to generate the populations.

In Garcia et al. (2013), the solution approaches based on ILS (Vansteenwegen et al., 2009) are adapted to deal with the Time Dependent Team OP with Time Windows (TDTOPTW), considering multiple paths (or multiple day visits) instead of only one. The city of San Sebastian is selected as a case study.

Gavalas, Konstantopoulos, Mastakas, Pantziou, and Vathis (2014b) use the TDTOPTW to model the TTDP. Two cluster-based algorithms, the Time Dependent CSCRoutes (TDCSCRoutes) and the SlackCSCRoutes, are proposed, which extend the previous algorithm CSCRoutes (Gavalas et al., 2013). The algorithms employ an insertion step that consider the travel time between two nodes and the waiting time (for public transport) which depends on the time the user arrives at the start node. TDCSCRoutes selects a node to be inserted based on the insertion cost, while SlackCSCRoutes involves a more global criterion as it considers the effect of the insertion on the whole path. Four algorithms, TDCSCRoutes, SlackCSCRoutes, AvgCSCRoutes and AverageILS, are tested using the General Transit Feed Specification (GTFS) data of the transit network deployed on Athens (Greece) accessible from http://www2.aegean.gr/dgavalas/public/tdtoptw_instances/index.html. AverageILS is based on the standard ILS algorithm (Vansteenwegen et al., 2009) and the average travel time approach (Garcia et al., 2013). AvgCSCRoutes uses CSCRoutes (Gavalas et al., 2013) to construct paths based on

Table 7
Papers on TDOP.

Reference	Problem	Algorithm	Application
Li (2012)	TDOP	Dynamic Programming	Transportation network
Verbeeck et al. (2014a)	TDOP	Ant Colony System	-
Gunawan et al. (2014)	TDOP	Iterated Local Search	Theme park navigation problem
Garcia et al. (2010)	TDOPTW	Hybrid Iterated Local Search	Personalized Electronic Tourist Guides
Abbaspour and Samadzadegan (2011)	TDOPTW	Adapted Genetic Algorithm	Tourist Trip Design Problem
Garcia et al. (2013)	TDTOPTW	Hybrid Iterated Local Search	Personalised Electronic Tourist guides
Gavalas et al. (2014b)	TDTOPTW	Time Dependent CSCRoutes and the SlackCSCRoutes	Tourist Trip Design Problem

pre-computed average travel times. Other applications of the TDOPTW related to the TTDP will be described further in Section 4.

2.4. Discussion

Section 2 illustrates that the TOP is still widely studied, as summarized in Table 1. One reason could be due to unknown optimal solutions of several benchmark instances. Another reason is that the TOP appears more as a model in practical problems. New and larger benchmark instances for the TOP are introduced (Dang et al., 2013b). This also provides room for research opportunities.

Some researchers focus on solving the TOPTW. A lot of new best known solutions of benchmark instances have been discovered, as summarized in Tables 4 and 5. Further improving the results of the current TOPTW benchmark instances becomes difficult. Probably it is useful to launch new benchmark instances with more than 300 nodes and/or more than 10 paths, where it is not possible to visit all nodes (which is the case in the current instances with more than 4 paths). Other new benchmark instances could consider clustered nodes or narrow or wide time windows. On the other hand, the TDOP, together with its variants, TDOPTW and TDTOPTW, is still not widely studied. Nevertheless, being able to model congestion issues becomes more and more a (minimal) requirement for solution techniques before they can be implemented in practice.

In general, most of the algorithms proposed for solving the classical (T)OP and its variants are based on ILS. Due to the increasing link with practical applications, only a few exact algorithms have been developed.

3. Extended Orienteering Problems

In this section, we discuss relevant and recent extensions of the OP. Many researchers have solved different problems by formulating them as variants of the OP. Some variants are already covered in previous surveys: e.g. Multi-objective OP (Schilde, Doerner, Hartl, & Kiechle, 2009), Multi-period OP (Tricoire, Romauch, Doerner, & Hartl, 2010), OP with hotel selection (Divsalar, Vansteenwegen, & Cattrysse, 2013; Divsalar, Vansteenwegen, Chitsaz, Sörensen, & Cattrysse, 2014a; Divsalar, Vansteenwegen, Sorensen, & Cattrysse, 2014b); therefore, these will not be discussed here again.

For some other variants such as Stochastic OP, Generalized OP and Arc OP, more papers have been published since the previous surveys and these more recent papers will be discussed here. Finally, a number of new OP variants, not surveyed before, such as the Multi-agent OP and others, will be discussed. These variants have been applied to different applications of the OP, and the details of the applications are described in Section 4.

3.1. Stochastic Orienteering Problem

In situations where congestion may occur, the travel times between nodes are not only time-dependent, but also difficult or even impossible to predict in a deterministic way. Therefore, OP variants with stochastic aspects have recently received more attention recently. Ilhan, Iravani, and Daskin (2008) are the first to introduce uncertainties in the collected scores. They discuss the OP with Stochastic Profits (OPSP) as a variant of the OP. The objective of the OPSP is to maximize the probability that the total collected score (profit) will be greater than a predefined target value.

In the OPSP, the profits associated with the nodes are stochastic with a known distribution. Each node $i \in N$ (excluding the start and end nodes) has a Normally distributed random profit $\tilde{S}_i$ with mean μ_i and σ_i^2. Given a target profit level, K, the objective function of the OPSP is to maximize the probability that the sum of the profits associated with the selected nodes is greater than or equal to K without violating the time budget constraint. The objective function is formulated as follows:

$$\text{Maximize } P\left(\sum_{i=2}^{|N|-1}\sum_{j=2}^{|N|}\tilde{S}_i X_{ij} \geq K\right) \tag{15}$$

There are other uncertainties or stochastic aspects that have been studied, especially related to the travel and service times, such as the OP with Stochastic Travel and Service times (OPSTS), the Dynamic Stochastic OP (DSOP), the OP with Stochastic Weights (OPSW) and the Stochastic OPTW (SOPTW). The differences among them, the benchmark instances and solution approaches for these variants are discussed in the next subsections.

3.1.1. Benchmark instances

Campbell, Gendreau, and Thomas (2011) introduce benchmark OPSTS instances. The instances are generated based on the benchmark OP instances (Chao et al., 1996a; Tsiligirides, 1984) by including the distributions of the travel times and some penalty values. These penalties are required in case nodes were scheduled to be visited, but cannot be reached due to the time budget.

Evers, Glorie, van der Ster, Barros, and Monsuur (2014) generate instances for the OPSW by modifying some benchmark OP instances (Chao et al., 1996b; Tsiligirides, 1984). Zhang, Ohlmann, and Thomas (2014) derive the SOPTW instances from Solomon's VRPTW instances (Solomon, 1987). Take note that most of the instances are not available online, except the benchmark instances by Zhang et al. (2014) which are available at http://ir.uiowa.edu/tippie_pubs/61/. However, the details of how to modify known instances can be found in the discussed papers.

3.1.2. Solution approaches

Table 8 presents an overview of the variants of the SOP including the characteristics, proposed algorithms and applications. All of them only focus on generating a single path.

Ilhan et al. (2008) introduce an exact algorithm that solves the problem without sub-tour elimination constraints and imposes those constraints only to eliminate sub-tours, if any appears. The experiments are conducted on four different problem sets. The Value of the Stochastic Solution (VSS) which represents the difference between the optimal stochastic objective value and that of the deterministic OP with expected profits is high and the stochastic and deterministic solutions are quite different from each other. They also propose a bi-objective Genetic Algorithm (GA) to tackle larger instances of the OPSP. The bi-objective GA tries to maximize the mean profit and to minimize the variance of the profit at the same time. It is concluded that the bi-objective GA is an alternative solution method for certain routing problems with non-linear objective functions.

Campbell et al. (2011) study another variant of the OP, namely the OP with Stochastic Travel and Service times (OPSTS). A penalty is incurred if customers are not reached before the pre-defined deadline. A variant of VNS is proposed for the OPSTS. The performance of VNS is compared with that of dynamic programming (DP). VNS is able to obtain good solutions within a few seconds, while DP grows exponentially in computational time with increasing deadlines.

Papapanagiotou, Montemanni, and Gambardella (2014) also study the OPSTS. The focus is to approximate the objective function more efficiently and to minimize loss in accuracy by implementing Monte Carlo sampling and hybrid methods combining Monte Carlo sampling and an analytical solution. The comparison is done with the analytical method used by Campbell et al. (2011). The hybrid methods can offer a reasonable approximation of the objective function in only a fraction of the time.

Lau, Yeoh, Varakantham, Nguyen, and Chen (2012) extend the OPSTS by considering dependencies between travel times and the

Table 8
Papers on SOP.

Reference	Problem	Characteristic	Algorithm	Application
Ilhan et al. (2008)	OPSP	Stochastic profits / scores	Exact Solution Algorithm and bi-objective Genetic Algorithm	Logistic problem
Campbell et al. (2011)	OPSTS	Stochastic travel and service times	Variable Neighborhood Search	Logistic problem
Papapanagiotou et al. (2014)	OPSTS	Stochastic travel and service times	Monte Carlo sampling and Hybrid Monte Carlo sampling and an analytical solution	Logistic problem
Lau et al. (2012)	DSOP	Stochastic time-dependent travel times	Hybrid Variable Neighborhood Search and Simulated Annealing	Theme park navigation problem
Varakantham and Kumar (2013)	DSOP	Stochastic time-dependent travel times	Mixed Integer Linear Programming-Sample Average Approximation	Theme park navigation problem
Evers et al. (2014)	OPSW	Stochastic travel and service times	Sample Average Approximation and OPSW heuristic	Logistic problem
Zhang et al. (2014)	SOPTW	Stochastic waiting time	Variable Neighborhood Search	Sales representative planning problem

risk preference of the user. The problem is called the Dynamic Stochastic OP (DSOP). The objective function is to maximize the sum of expected scores (minus penalties) from nodes in the sequence. Travel times are modeled with random variables that follow a given time-varying distribution. The risk profile of the user is measured as the probability of completing the path within the time budget. They introduce a local search algorithm that combines VNS and SA. The experiments are conducted on modified instances (Campbell et al., 2011) and a real-world instance from a theme park navigation problem. The hybrid approach is able to improve the initial solution generated by a greedy insertion algorithm up to 30 percent.

Varakantham and Kumar (2013) extends of the work of Lau et al. (2012). Although the proposed local search approach is scalable, it is ad-hoc and it does not provide any a priori or posteriori guarantees that the optimal solution is found. An optimization based approach that employs ideas from the sample average approximation technique, namely Mixed Integer Linear Programming-Sample Average Approximation (MILP-SAA), is proposed. The approach is compared with the local search approach (Lau et al., 2012) on a set of modified benchmark instances (Campbell et al., 2011) and a real theme park navigation problem. It is concluded that MILP-SAA outperforms the local search approach over synthetic problems and a real-world problem.

Evers et al. (2014) focus on another variant of the SOP, namely the OP with Stochastic Weights (OPSW). Here, weights are associated with travel costs, travel time (including service time) or fuel consumption on arcs. Some factors such as weather circumstances and congestion may affect the uncertainty of the weights. A two-stage recourse model is introduced to formulate the OPSW. The first stage decision is to find a path. In the second stage, the weight realizations are revealed and recourse costs are imposed. Sample Average Approximation (SAA) using Monte Carlo simulation is used to solve the problem. The idea is to generate candidate solutions by a random sample. SAA can only solve small instances within reasonable computation time. Therefore, another heuristic approach is proposed based on a randomization concept (Tsiligirides, 1984) and a score measure (Golden, Wang, & Liu, 1988), namely OPSW heuristic. Computational results show the benefit of using the proposed approaches. SAA takes more computation time than the OPSW heuristic. The OPSW heuristic produces a higher expected profit compared to applying a heuristic for the deterministic OP in an uncertain environment.

Zhang et al. (2014) extend the SOP by considering the time window constraints, namely the Stochastic OP with Time Windows (SOPTW). The waiting time is modeled as a random variable dependent on the arrival time and the queue length upon arrival. The main objective is to construct an a priori path that maximizes the expected score collected on a given day. During the execution of an a priori path, two recourse actions are considered. The first recourse determines whether a node should be skipped based on the arrival time at that particular node. The second one determines how long we should wait after arriving and observing a queue. A VNS heuristic is proposed for solving the benchmark SOPTW instances. The algorithm is a variant of the one used by Campbell et al. (2011) by including the time window constraint. The lower and upper bounds for the SOPTW are obtained by solving the deterministic OPTW using a dynamic programming approach. It is concluded that the SOPTW solutions are 9.2 percent above the lower bound and about 26.3 percent below the upper bound. The SOPTW solution improves the deterministic approach solution by 9.2 percent on average.

3.2. Generalized Orienteering Problem

The Generalized OP (GOP) is a generalized version of the OP in which each node is assigned a set of scores with respect to a set of attributes (Geem, Tseng, & Park, 2005). The overall objective function in the GOP is a non-linear function of these attribute scores (Wang, Golden, & Wasil, 2008). This is different from the standard OP where the objective function value is the sum of individual scores from selected nodes. We only present the objective function of the GOP since the constraints are the same with the ones of the OP (Section 2.1).

Each node $i \in N$ has a score vector $S(i) = (S_1(i), S_2(i), \ldots, S_g(i))$, where g is the number of independent attributes. $S_g(i)$ represents the score of node i with respect to attribute g. The objective function of the GOP, $\bar{Z}$, is to maximize the total score of a path P, as presented below:

$$\text{Maximize } \bar{Z} = \sum_{j=1}^{g} W_j \left[\{\sum_{i \in P} [S_j(i)]^k\}^{1/k} \right] \quad (16)$$

Objective function (16) requires a weight W_j for each attribute j, such that $\sum_{j=1}^{g} W_j = 1$. Let k be a non-negative exponent. The OP is a special case of the GOP if we set $k = 1$ and $g = 1$. As k approaches infinity, $\bar{Z}$ approaches Z, where

$$\text{Maximize } Z = \sum_{j=1}^{g} W_j \left\{ max_{i \in P}(S_j(i)) \right\} \quad (17)$$

3.2.1. Benchmark instances

Wang, Sun, and Golden (1996) introduce an instance in the context of 27 cities in China. It includes the longitudes, latitudes and four scores which represent natural beauty, historical significance, cultural-educational attractions and business opportu-

Table 9
Papers on GOP.

Reference	Problem	Algorithm	Application
Geem et al. (2005)	GOP	Harmony Search	Tourist trip design problem
Wang et al. (2008)	GOP	Genetic Algorithm	Tourist trip design problem
Silberholz and Golden (2010)	GOP	Two-parameter iterative algorithm	Tourist trip design problem
Pietz and Royset (2013)	GOP-RDR	Hybrid Branch-and-bound and four-phase heuristic	Smuggler search problem

nities for each city. The characteristics of this instance can be found in Geem et al. (2005) and Wang et al. (2008). The instance is also available at http://www.terpconnect.umd.edu/~bgolden/vrp_data.htm.

3.2.2. Solution approaches

Table 9 summarizes the proposed algorithms and the applications of the GOP and its variant, the GOP with Resource Dependent Rewards (GOP-RDR).

Geem et al. (2005) introduce a nature-inspired algorithm, namely Harmony Search (HS). The idea of HS is based on an analogy with a music improvisation process where each music player improvises the pitch of his instrument to obtain better harmony (Geem, Kim, & Loganathan, 2001). Three major behaviors of music players: MEMORY CONSIDERATION, PITCH ADJUSTMENT and RANDOM CHOICE, are translated into the context of the GOP. MEMORY CONSIDERATION refers to the selection of any node from nodes stored in the Harmony Matrix (HM). PITCH ADJUSTMENT focuses on selecting the nearest node as the next node to be visited, and RANDOM CHOICE concerns on choosing one node from all possible nodes. HS is applied to find the best tour for the benchmark instance (Wang et al., 1996). The experiments are conducted with different weight values for attributes. The results obtained are compared with those of the Artificial Neural Network (ANN) approach (Wang et al., 1996). It is concluded that HS performs better than ANN does.

Wang et al. (2008) propose a Genetic Algorithm (GA) to solve the same problem. The GA consists of standard operations, such as CROSSOVER, 2-OPT and proportional fitness selection. The results are compared with the ones of the ANN approach. In terms of the solution quality, GA can produce comparable results with less computational time. We observe that GA outperforms HS in terms of the total distance travelled and the total score.

Silberholz and Golden (2010) design a two-parameter iterative algorithm. The initial solution is generated by appending nodes to the end of the path. A set of nodes are first selected randomly and followed by adding the one at the end of the current path that minimizes the total distance. This is continued until either all nodes are added to the path or the length of the path does exceed the limit. After generating the initial solution, two standard local search procedures are applied iteratively: 2-OPT, INSERT and REMOVE. The results obtained are compared to previous results (Geem et al., 2005; Wang et al., 2008). The two-parameter iterative algorithm outperforms HS, ANN and GA in terms of solution quality.

Pietz and Royset (2013) define another extension of the GOP, namely the GOP with Resource Dependent Rewards (GOP-RDR). This problem generalizes the OP by allowing node rewards and arc length to vary based on the amount of the resources spent at each node. The mathematical model and a branch-and-bound algorithm are proposed. A five-phase heuristic is introduced to solve the problem. It combines a branch-and-bound algorithm and a four-phase heuristic proposed by Ramesh and Brown (1991). The algorithm is implemented on 100 randomly generated problem instances. The comparison is done by solving the mathematical model directly using Mixed Integer Non Linear Programming (MINLP) solvers. It is concluded that for problems with seven or more nodes, the proposed algorithm outperforms the standard MINLP solver.

3.3. Other variants

In this section, we summarize the recent research about other variants of the OP, such as the Arc OP, the Capacitated Team OP, the Multi-agent OP, the Multi-Period OP with Multiple Time Windows and others.

Arc Orienteering Problem

The **Arc OP** (AOP) is considered as an arc variant of the OP where the focus lies on visiting arcs between nodes while other variants focus on visiting nodes (Souffriau, Vansteenwegen, Vanden Berghe, & Van Oudheusden, 2011). Archetti and Speranza (2014) provide a survey related to the AOP and the extension of the AOP for the multiple vehicle case, namely the **Team Orienteering Arc Routing Problem** (TOARP). We briefly extend the survey by including most recent works related to both AOP and TOARP, as shown in Table 10.

Souffriau et al. (2011) introduce a real-life test set which is generated from the cycle network of the province of East Flanders (Belgium), namely *FlandersCycle*. The 50 benchmark instances are consolidated in http://www.mech.kuleuven.be/en/cib/op/. Mousselly-Sergieh et al. (2014) present a dataset of geotagged photos on a world-wide scale. This dataset contains a sample of millions of photos crawled from Flickr. The data can be found at https://drive.google.com/folderview?id=0B-mRR4rjwHPOQUJ1d0x5aHVHVWM&usp=sharing. This dataset is used to create benchmark AOP instances by Lu and Shahabi (2015).

Verbeeck, Vansteenwegen, and Aghezzaf (2014b) also study the AOP, in the context of cycle trip planning. They develop an exact branch-and-cut algorithm and ILS-approach to solve Cycle Trip Planning Problem instances, AOP instances and modified AOP instances. The instances used in the experiments are all based on the FlandersCycle instances of Souffriau et al. (2011).

Gavalas, Konstantopoulos, Mastakas, Pantziou, and Vathis (2015b) propose approximation algorithms for the AOP in directed and undirected graphs. A polylogarithmic approximation algorithm and a $(6+\epsilon+o(1))$-approximation algorithm are introduced for the AOP in directed and undirected graphs, respectively. Another algorithm, a $(4+\epsilon)$-approximation algorithm, is proposed for the special case of the AOP with arcs of unit profit, namely the unweighted AOP (UAOP).

Lu and Shahabi (2015) focus on solving the AOP on a large real road network. The road network is treated as a spatial network. They propose three ILS-based algorithms by utilizing the spatial database techniques: ellipse pruning and spatial indexing. The proposed algorithms are compared with two algorithms: GRASP (Souffriau et al., 2011) and ILS (Verbeeck et al., 2014b) in solving two real datasets (Mousselly-Sergieh et al., 2014; Souffriau et al., 2011). Experiment results show the efficiency and accuracy of the proposed algorithms within very short computational times.

Archetti, Speranza, Corberán, Sanchis, and Plana (2014b) generate a set of instances for the TOARP which is based on a set for the undirected rural postman problem (RPP). In total, there are 69 instances with different percentage of regular and potential nodes. The instances can be accessed at http://www.uv.es/corberan. Archetti, Corberán, Plana, Sanchis, and Speranza (2015) introduce a new class of randomly generated instances. Archetti et al. (2014b) and Archetti et al. (2015) propose a

Table 10
Recent papers on AOP.

Reference	Problem	Algorithm	Application
Verbeeck et al. (2014b)	AOP	Iterated Local Search and Branch-and-cut algorithm	Cycle trip planning problem
Gavalas et al. (2015b)	AOP	Approximation algorithms	-
Lu and Shahabi (2015)	AOP	Modified Iterated Local Search	Tourist trip design problem
Archetti et al. (2014b)	TOARP	Branch-and-cut algorithm	Logistic problem
Archetti et al. (2015)	TOARP	Hybrid Tabu Search and diversification phase with the exact solution of ILP models	Logistic problem
Archetti et al. (2016)	OARP	Branch-and-cut algorithm	Logistic problem

Table 11
Papers on CTOP.

Reference	Problem	Characteristic	Algorithm	Application
Archetti et al. (2009)	CTOP	Associated demand for each node and the capacity constraint	Tabu Search and Variable Neighborhood Search	Logistic problem
Archetti et al. (2013b)	CTOP	Associated demand for each node and the capacity constraint	Branch-and-price algorithm	Logistic problem
Tarantilis et al. (2013)	CTOP	Associated demand for each node and the capacity constraint	Hybrid Tabu Search, a Filter-and-Fan method and Variable Neighborhood Descent	Logistic problem
Archetti et al. (2013a)	CTOP-IS	Associated partial demand for each node and the capacity constraint	Branch-and-price algorithm	Logistic problem
Archetti et al. (2014a)	SDCTOP	Split demand for each node and the capacity constraint	Branch-and-price algorithm and Hybrid Tabu Search	Logistic problem
Wang et al. (2014)	SDCTOP-MDA	Split demand with minimum delivery amounts	Worst-case analysis	-

branch-and-cut algorithm and a matheuristic, respectively. Archetti, Corberán, Plana, Sanchis, and Speranza (2016) introduce **the Orienteering Arc Routing Problem** (OARP) as a simplified version of the TOARP with a single vehicle and propose a branch-and-cut algorithm. A new set of larger instances can be accessed at http://www.uv.es/corberan as well.

Capacitated Team Orienteering Problem

The **Capacitated TOP** (CTOP) is another variant of the TOP where each node is associated with a demand and a score (Archetti, Feillet, Hertz, & Speranza, 2009). The main objective is to determine a path for each available vehicle in order to maximize the total score, without violating the capacity and time budget of each vehicle. Table 11 presents an overview of different variants of the CTOP.

Two variants of TS and VNS are proposed to solve benchmark instances (http://tarantilis.dmst.aueb.gr/docs/). The results show that the heuristics obtain very good results within a reasonable amount of time. Archetti, Bianchessi, and Speranza (2013b) propose a branch-and-price algorithm or IP column generation solution algorithm. It outperforms the algorithm proposed by Archetti et al. (2009) and solves several unsolved benchmark instances to optimality.

Tarantilis, Stavropoulou, and Repoussis (2013) also study the CTOP by proposing a hierarchical bi-level search framework, namely a Bi-level Filter-and-Fan method. At the upper master level, the objective is to select the subset of nodes by employing a Filter-and-Fan method. TS and an innovative Filter-and-Fan search (Rego & Glover, 2010) are iteratively applied. At the lower level that corresponds with the vehicle routing problem, the objective is to minimize the total travel distance via a Variable Neighborhood Descent algorithm. The proposed algorithm is tested on benchmark instances (Archetti et al., 2009) and a new set of large scale benchmark CTOP instances. All instances can be downloaded from http://tarantilis.dmst.aueb.gr/docs/. Experimental results show the efficiency and effectiveness of a Bi-level Filter-and-Fan method. It is able to improve 18 best known solutions. On a new set of instances, it produces high quality solutions with reasonable computational times.

Archetti, Bianchessi, and Speranza (2013a) extend the CTOP by relaxing the assumption that a served customer must be completely served, namely the **CTOP with Incomplete Service** (CTOP-IS). It is proven that the profit collected by the CTOP-IS could be as large as twice the one collected by the CTOP. Archetti, Bianchessi, and Speranza (2014a) extend the CTOP by allowing a node to be served by more than one path. In the context of VRP, each path is associated with each vehicle. The problem is named as the **Split Delivery CTOP** (SDCTOP). By splitting deliveries, the cost of paths and the number of vehicles used are reduced (Archetti, Hertz, & Speranza, 2006). Two heuristics are proposed to solve the problem. The first heuristic makes use of the columns generated by branch-and-price heuristic solutions. A mixed ILP is solved on subsets of promising columns. The second heuristic is a hybridization of TS and an improvement phase where a mixed ILP is solved to intensify the search. Experiments are conducted on benchmark CTOP instances (Archetti et al., 2009) and a new set of benchmark instances. It is concluded that the collected score increases due to split deliveries. The branch and price algorithm is able to solve several instances optimally. The hybrid TS also produces high quality solutions.

Wang, Golden, and Gulczynski (2014) extend the SDCTOP by introducing another problem, namely the **Split Delivery TOP with Minimum Delivery Amounts** (SDCTOP-MDA). A worst-case analysis of the problem for determining tight bounds on the maximum possible score increase is performed. It is concluded that by allowing split deliveries with a minimum delivery fraction strictly less than $\frac{1}{2}$, the collected score can be doubled. If the minimum delivery fraction is set to $\frac{1}{2}$, the score can increase with up to 50 percent.

Finally, we summarize other variants of the OP together with their characteristics, algorithms and applications in Table 12.

Table 12
Papers on other variants of the OP.

Reference	Problem	Characteristic	Algorithm	Application
Erdoğan and Laporte (2013)	OPVP	Multiple visits and staying a predefined amount of time at nodes to collect a higher score	Branch-and-cut algorithm	Logistic problem
Afsar and Labadie (2013)	DPTOP	Decreasing function of time for each node's score	Dantzig–Wolfe decomposition and Evolutionary Local Search	Repairing-maintenance problem
Angelelli et al. (2014)	COP	Clustered nodes	Branch-and-cut algorithm and Tabu Search	Logistic problem
Yu et al. (2014)	CorOP	Quadratic score function	Exact algorithm	Robot problem
Van der Merwe et al. (2014)	COPTW	Cooperative vehicles	Exact algorithm	Wildfire asset protection
Chen et al. (2014b)	MOPTCC	Multiple agents and limited capacity	Exact algorithm and the Sampled Fictitious Play algorithm	Crowd control problem
Varakantham et al. (2015)	SeOP	Selfish agents with budget constraints	Incremental and iterative master-slave decomposition approach	Theme park problem
Tricoire et al. (2010)	MuPOPTW	Multiple time periods and multiple time windows	Variable Neighborhood Search	Sales representative planning problem
Souffriau et al. (2013)	MCTOPMTW	Multiple attributes and multiple time windows	Hybrid Greedy Randomized Adaptive Search Procedure and Iterated Local Search	Tourist trip design problem
Lin and Yu (2015)	MCTOPMTW	Multiple attributes and multiple time windows	Hybrid Simulated Annealing with restart strategy	Tourist trip design problem
Salazar-Aguilar et al. (2014)	MDTOP	Multiple districts with mandatory and optional nodes	Adaptive Large Neighborhood Search	Crowdsourcing problem

Orienteering Problem with variable profits

Erdoğan and Laporte (2013) study a variant of the OP, namely the **OP with Variable Profits** (OPVP). The underlying assumption is that the collection of scores at a particular node require either a number of discrete passes or a continuous amount of time to be spent at that node. The collected score on node i depends on an associated collection parameter $\alpha_i \in [0, 1]$. Both discrete and continuous models are formulated as a linear integer programming model and a non-linear integer programming model, respectively. The experiments are conducted on instances adapted from the TSPLIB (Reinelt, 1991). It is shown that the discrete model can be solved for instances with up to 200 nodes within 2 hours of computational time. On the other hand, the continuous model requires more computation time already for instances with 75 nodes.

Afsar and Labadie (2013) extend the TOP into the **TOP with Decreasing Profits** (DPTOP). The profit of each node is a decreasing function of time. Due to the complexity of the problem, the Column Generation approach (CG) is introduced to reformulate and calculate the lower and upper bounds of the initial DPTOP integer programming model. Evolutionary Local Search (ELS) is also proposed to solve the problem. TOP benchmark instances (Chao et al., 1996b) are modified by adding the variable profit for the experiments. Almost all instances can be solved optimally by CG with the cost of computational time, while the ELS is less competitive in terms of the quality of solutions.

Clustered Orienteering Problem

The **Clustered OP** (COP) is a generalization of the OP when nodes are clustered in groups (Angelelli, Archetti, & Vindigni, 2014). A score is associated with each group and can only be gained if all nodes in one particular group are served. One example of an application of the COP is the distribution of mass products to all retailers belonging to one particular chain. The underlying assumption is that if a carrier agrees to serve a particular chain, he or she has to serve all retailers in that chain.

Two approaches based on a branch-and-cut algorithm and TS are proposed to solve the COP. The branch-and-cut algorithm, namely COP-CUT, relaxes sub-tour elimination constraints and will only insert them once violated. A Tabu Search algorithm, namely COP-TABU, is proposed to solve larger instances. The neighborhoods generated are mainly based on two standard operations: INSERT and REMOVE operations. Three variants of COP-TABU are considered with the emphasis on how to rerun COP-TABU after a certain number of iterations. Since there are no benchmark instances for the COP, TSP benchmark instances from the TSPLIB with some modifications are used to test the proposed approaches. The instances can be found at the following URL: http://or-brescia.unibs.it/. The branch-and-cut algorithm is only able to solve small to medium size instances. COP-TABU can provide high quality solutions within a very short computational time. It is concluded that the performance of the proposed algorithm does not depend on the number of groups.

Correlated Orienteering Problem

Yu, Schwager, and Rus (2014) study the **Correlated OP** (CorOP), a quadratic extension of the OP. The objective function consists of two components: a total of collected score from visited nodes and a quadratic score function that captures spatial correlations among nodes. The scores to be collected at nodes are frequently correlated between nodes that are close to each other. The problem addressed is related to single-robot and multi-robot tours. Robots must visit nodes in order to maximize the objective function but they have limited time budget. The CorOP for single and multiple tours are formulated as Mixed Integer Quadratic Programming (MIQP) models. The limitation of solving networks with multiple robots and hundreds of nodes quickly is highlighted.

Cooperative Orienteering Problem

Van der Merwe, Minas, Ozlen, and Hearne (2014) present a generalization of the TOPTW where a certain number of resources or vehicles are required to collect the score from a particular node. This problem is known as the **Cooperative OP with Time Windows** (COPTW). It is also required that resources have to arrive before service may commence. A two-index vehicle flow formulation is presented. Modified instances (Vansteenwegen et al., 2009) by adding a column for the resource requirements are used to demonstrate the computational time for solving the mathematical model using CPLEX 12.6. Most problems with 20 nodes or fewer and a real case study can be solved optimally. However, for larger number of nodes, the development of fast heuristic approaches are considered as further work.

Multi-agent Orienteering Problem

Multi-agent OP (MOP) is another new variant of the OP. Unlike the classical setting of the OP, which is concerned with the path planning for a team of agents in a centralized fashion, the problem is treated as a multi-agent planning problem where individual agents are self-interested and will interact with each other when they arrive at the same nodes simultaneously. Chen, Cheng, and Lau (2014b) study the **MOP with Time-dependent Capacity Constraints** (MOPTCC). Due to the capacity constraint, each node can only receive a limited number of agents at the same time. If more agents are present, all agents will have to wait due to some extra queueing time. Therefore, the main focus is to identify a Nash equilibrium where individual agents cannot improve their current utilities by deviation. The problem is formulated as an integer programming model and a game-theoretic formulation.

Chen et al. (2014b) generate random instances for the MOPTCC. The number of agents and nodes are limited to 8 and 10, respectively. They propose two solution approaches: a centralized approach with Integer Linear Programming (ILP) that computes the exact global solution and a variant of the Sampled Fictitious Play (SFP) algorithm (Lambert III, Epelman, & Smith, 2005) that efficiently identifies equilibrium solutions. However, the first approach does not scale well and can only solve very small instances. The computational experiments show the ability of finding the equilibrium solutions in randomly generated instances.

Varakantham, Mostafa, Fu, and Lau (2015) model the problem of crowd congestion at certain venues like theme parks, museums and world expos as a variant of the OP. The main issue is how to provide route guidance to multiple selfish users (with budget constraints) moving through the venue simultaneously. The **Selfish OP** (SeOP) that combines the OP and Selfish Routing (SR) is introduced to represent these settings. SR is a game between selfish agents looking for minimum latency paths from source to destination along edges of a network available to all agents. Thus, SeOP is the MOP where agents have selfish interests and individual budget constraints. As with Selfish Routing, Nash Equilibrium as the solution concept in solving SeOP is employed. A direct mathematical program formulation to find a Nash equilibrium in SeOP is not scalable because the number of constraints is quadratic in the number of paths, which itself is an exponential quantity. To address this issue, they provide a compact non-pairwise formulation with linear number of constraints in the number of paths to enforce the equilibrium condition. DIRECT, an incremental and iterative master-slave decomposition approach to compute an approximate equilibrium solution, is also introduced. Similar to existing flow based approaches, DIRECT is scale-invariant in the number of agents. A theoretical discussion of the approximation quality and experimental results clearly show that the non-pairwise formulation achieves the same solution quality as the pairwise one using a fraction of the number of constraints and the master-slave decomposition achieves solutions with an adjustable approximation gap using a fraction of the full path set.

Orienteering Problem with multiple aspects

Tricoire et al. (2010) study the **Multi-Period OP with Multiple Time Windows** (MuPOPTW) by combining (T)OPTW constraints, standard vehicle routine problems and a real industrial case. Two heuristics are proposed: a constructive heuristic based on the best insertion heuristic (Solomon, 1987) and VNS. The experiments are conducted on 60 different problem instances provided by the industrial partner that can be accessed in http://prolog.univie.ac.at/research/OP/. The VNS is able to provide good solutions within a reasonable amount of time. The heuristics are also compared with the best known methods of (T)OPTW and (T)OP problems in solving benchmark instances. In general, the VNS provides good solutions although it requires more computational time.

The **Multi-Constraint TOP with Multiple Time Windows** (MCTOPMTW) is introduced by Souffriau et al. (2013). This problem is linked to the TTDP where nodes may have one or more time windows. The additional constraints added into the problem are budget limitations for entrance fees and "max-*n* type constraints" for each day or for the whole trip. In addition, the drawback of the TOPTW as a model for the TTDP is that it assumes the same single time window for each day. The MCTOPMTW model allows different time windows on different days and more than one time window per day. A fast and effective algorithm based on the hybridization of ILS (Vansteenwegen et al., 2009) and GRASP, namely GRILS, is proposed to tackle the problem. The algorithm is tested on four different sets of instances. Each set of instances represents a different variant of the OP: TOPTW (Vansteenwegen et al., 2009), SVRPTW (Boussier et al., 2007), MCTOPTW (Garcia et al., 2013) and difficult-MCTOPTW. These benchmark instances can be found in http://www.mech.kuleuven.be/en/cib/op/. The last set of instances is generated from the extended TOPTW instances with the additional constraints mentioned above. For the TOPTW instances, although GRILS is not the best performer compared with other methods (Montemanni & Gambardella, 2009; Tricoire et al., 2010; Vansteenwegen et al., 2009), it is claimed that GRILS is suited for solving problems in real-time and finding three new best solutions. Moreover, GRILS outperforms ILS (Garcia et al., 2013) for solving both SVRPTW and MCTOPTW instances. The results of GRILS in solving difficult-MCTOPTW instances are very good. The average score gap with known optimal solutions is around 5.19 percent, using 1.5 seconds of computational time.

Lin and Yu (2015) propose two versions of SA with restart strategy (SA-RS) for solving the MCTOPMTW. The first version (SA-RS_{BF}) uses a Boltzmann function in order to accept a worse solution while the second one (SA-RS_{CF}) is based on the acceptance probability determined by a Cauchy function. The performance of SA-RS are compared with GRILS (Souffriau et al., 2013) and SA without restart strategy (SA_{BF} and SA_{CF}) in solving benchmark instances. The computational results confirm the superiority of SA-RS_{CF}. It is able to find 6 new best known solutions.

Salazar-Aguilar, Langevin, and Laporte (2014) introduce an extension of the TOP by considering the multi-district aspect, a set of mandatory and optional tasks located in several districts and some incompatible tasks which cannot be carried out during the same day. The problem is called the **Multi-District TOP** (MDTOP). It is required to perform all mandatory tasks over the planning horizon, while the optional tasks are only executed if time permits. The planning horizon is distributed among districts during the planning stage. The objective is to maximize the total collected score related to each optional task. The problem is formulated as a Mixed ILP model. Due to the complexity of the problem, an Adaptive Large Neighborhood Search (ALNS) metaheuristic is proposed. A large set of artificial instances are generated by using different numbers of districts, the number of incompatible tasks, the number of mandatory tasks and scores of optional tasks. The performance of the ALNS is very promising by considering the size and difficulty of the instances.

3.4. Discussion

In Section 3, we observe that many researchers have focused their research works beyond the classical OP by considering more complex characteristics, such as correlated nodes, non-linear objective functions, multiple time windows and so on. In terms of algorithms proposed, it is interesting to note that the idea of

Table 13
Papers on mobile crowdsourcing problem.

Reference	Algorithm	Related OP
Ludwig et al. (2009)	A*-like algorithm	OP and MPOPTW
Liao and Hsu (2013)	Dynamic Programming	OPTW
Chen et al. (2014a)	Iterated Local Search	TOP
Chen et al. (2015)	Lagrangian Relaxation	MuPOPTW

Table 14
Papers on TTDP.

Reference	Algorithm	Related OP
Souffriau et al. (2008)	Guided Local Search	OP and TOP
Sylejmani and Dika (2011)	Tabu Search	TOPTW
Sylejmani et al. (2014)	Simulated Annealing	MCTOPMTW
Herzog and Wörndl (2014)	Approximated Knapsack Problem algorithm	OP with budget constraints
Verbeeck et al. (2014b)	Branch-and-cut algorithm and Iterated Local Search	AOP
Malucelli et al. (2015)	Exact algorithm	MOP-ND
Yu et al. (2015)	Exact algorithm and heuristic based on arbitrarily optimal MIP model	OPVP
Gavalas et al. (2015a)	SlackRoutes Algorithm	TDOP

hybridizing exact algorithms and heuristics has become more popular over the past five years.

Furthermore, several sets of benchmark instances for new variants of the OP have been introduced, such as instances of the MuPOPTW, MVTOPMTW and MDTOP. This should encourage researchers to develop new and better algorithms for dealing with these instances. Finally, different real applications of the OP and its variants, such as logistic problems and crowdsourcing problems, have been studied. Therefore, we continue this paper by surveying, in Section 4, the latest applications modeled by the OP.

4. Orienteering Problem applications

In recent years, it is clear that variants of the OP are used more and more in order to model planning problems from practice (Vansteenwegen et al., 2011a). We present a summary of several practical applications, such as the mobile crowdsourcing problem, the Tourist Trip Design Problem (TTDP) and others. In this section, we focus on recent applications and how appropriate variants of the OP are used to model practical applications.

4.1. Mobile crowdsourcing problem

Howe (2008) defines crowdsourcing as an idea of outsourcing a task that is traditionally performed by an employee to a large group of people in the form of an open call. It refers to the detour planning problem, in which some companies outsource tasks to individuals who are willing to complete them for rewards. Each individual should then solve a variant of the OP in order to determine their contribution to the crowdsourcing. There are various applications of the crowdsourcing, for example, a mobile crowdsourcing application that computes the best detour paths for smartphone users who have some time to spare to perform some additional tasks for small rewards. For a comprehensive survey about the crowdsourcing problem itself, we can refer to Yuen, King, and Leung (2011). Table 13 summarizes different crowdsourcing papers including the proposed algorithms and relationship to the OP and its variants.

Ludwig, Zenker, and Schrader (2009) propose ROSE which is a mobile application that combines event recommendation and pedestrian navigation with (live) public transport support. It consists of three main parts: recommendation, route generation and navigation parts. In the context of the number of visited nodes, they differentiate two different scenarios: single destination and multiple destination modes. The former only concerns visiting a single node while the latter focuses on several nodes. They highlight that the multiple destination modes can be modeled as the OP. By taking public transport and time windows into account, the problem is extended to the Multi-Path OP with Time Windows (MPOPTW) (Garcia, Linaza, Arbelaitz, & Vansteenwegen, 2009).

Liao and Hsu (2013) formulate the crowdsourcing problem as a generalized version of the OPTW. They propose a crowdsourcing system for multimedia content gathering where requests must be performed by users at specific locations and time. The corresponding requesters could be police departments who need to collect evidence of crime scenes and people who need photos of memorable locations. In this problem, a single user (e.g. a smartphone user) may take some detour paths for getting additional rewards as long as he can reach his final destinations in time. A generalized OP-solution algorithm is designed in order to generate a detour path and to maximize the user's profit from actual geospatial traces from Flickr.

Chen et al. (2014a) investigate the problem of large-scale mobile crowdsourcing. The problem considers a large pool of crowdworkers to perform a variety of location-specific urban tasks. The assignment of tasks that consider the inherent paths of workers can be modeled as the TOP. The major difference lies on the node classification. They classify nodes into two categories: routine nodes that must be visited and task nodes that may or may not be visited. The objective is to maximize the total collected score from visited task nodes while respecting time budget constraints.

Chen, Cheng, Lau, and Misra (2015) extend the above problem by assuming that each user has a finite list of possible paths with a known probabilistic distribution. This problem is considered as the extension of the Multi-Period OP with Multiple Time Windows (MuPOPTW) (Tricoire et al., 2010). In the MuPOPTW context, sales representatives need to visit a list of mandatory customers on a regular basis, while optional customers located nearby should also be considered and integrated into the current paths. While one may view the set of mandatory customers as the routine nodes and non-mandatory customers as the task nodes. They add predefined visiting sequence for the mandatory customers which is not captured in the MuPOPTW.

4.2. Tourist Trip Design Problem

The Tourist Trip Design Problem (TTDP) is defined as a route-planning problem for tourists interested in visiting multiple Points Of Interest (POIs) (Vansteenwegen & Van Oudheusden, 2007). The most basic version of the TTDP corresponds to the OP. Gavalas et al. (2014a) provide a detailed survey of the TTDP. We present the most recent papers not mentioned in previous surveys. Table 14 summarizes those papers, including proposed algorithms and the relationship with the OP and its variants.

Souffriau, Vansteenwegen, Vertommen, Vanden Berghe, and Van Oudheusden (2008) study the problem of developing intelligent Mobile Tourist Guides (MTG). The attractiveness of each POI is assumed to be determined first. It depends on different fields of interest. The way of computing the score for each POI based on information retrieval techniques is discussed. The objective of the MTG is to maximize the collected score by visiting attractions during the limited available time. For a one day trip, this corresponds to the OP; for multiple day trips, this corresponds to the TOP.

Sylejmani and Dika (2011) also solve a TTDP. In their proposed mathematical model, tourists are not allowed to reach the POI before its opening time, which is slightly different from the classical TOPTW. They solve instances for the city of Vienna (Austria). In conclusion, they are able to produce a personal trip itinerary within reasonable computational times.

Sylejmani, Muhaxhiri, Dika, and Ahmedi (2014) extend the previous TTDP by modeling it as the Multi-Constraint TOP with Multiple Time Windows (MCTOPMTW). They propose an algorithm based on SA for solving this problem. The neighborhoods consider three operators, namely Insert, Swap and Shake. Several instances from fifty POIs in the city of Prishtina (Kosova) are selected for the experiments. Different scenarios of experiments are conducted in order to evaluate the performance of the proposed algorithm. The algorithm produces comparable results for all instances.

Herzog and Wörndl (2014) study another variant of the TTDP for an individual user where trips are composed by multiple regions, namely a composite trip. The problem of combining regions to a composite trip is part of the OP. The main objective is to select travel regions that maximize the value of the composite trip for the user while still respecting the limitation in terms of time and money. They present a travel recommendation which allows the user to specify his queries for composite trip recommendation.

Verbeeck et al. (2014b) introduce another variant of the TTDP that focuses on large cycling networks, namely the Cycle Trip Planning Problem (CTPP). This problem is considered as the application of the AOP. The directed graph consists of a set of nodes and a set of arcs. Each arc corresponds with a cost (e.g. distance or travel time) and a profit. There is no fixed starting node in this routing problem. The objective is to determine a closed path that maximizes the total collected score.

Malucelli, Giovannini, and Nonato (2015) study the problem of designing the most attractive itineraries for a single origin-destination pair for different classes of users. The problem is formalized as an integer programming model underlining common features with the OP and the Multicommodity Minimum Cost Flow with Network Design Problem, namely the Multi-commodity OP with Network Design (MOP-ND). The model is tested on real data for the Trebon region (Czech Republic). It is concluded that considering the preferences of different classes of users allows for higher quality itineraries.

Yu, Aslam, Karaman, and Rus (2015) propose the Optimal Tourist Problem that combines the problem of maximizing information collection efforts at POIs and minimizing the time spent on traveling between the set of discrete, spatially distributed POIs. Computational results illustrate that the proposed algorithm is applicable to generate a day tour of Istanbul over 20 POIs. This problem is considered as the application of the OP with Variable Profits.

Gavalas et al. (2015a) develop the web and mobile eCOMPASS applications using the metropolitan areas of Athens (Greece) and Berlin (Germany) as case studies. A metaheuristics based on a local search, namely the SlackRoutes, is employed for solving the TTDP.

We also include some papers that have been described in Section 3. Here, we only briefly mention the considered applications and their relation to the TTDP. Geem et al. (2005), Wang et al. (2008) and Silberholz and Golden (2010) solve a TTDP in the context of finding the best tour in the eastern part of China (Wang et al., 1996), modeled as the GOP. Lu and Shahabi (2015) focus on solving a TTDP in the context of finding the most scenic path on a large real road network, taken from *FlandersCycle* (Souffriau et al., 2011) and *LAFlickr* (Mousselly-Sergieh et al., 2014). *FlandersCycle* is based on a cycle network of East Flanders and *LAFlickr* uses the Los Angeles road network. Some other papers, such as Souffriau et al. (2013) and Lin and Yu (2015), only mention briefly possible applications to the TTDP.

4.3. Other applications

Table 15 summarizes other applications of variants of the OP. For most papers, the algorithms are already discussed in earlier sections.

Lau et al. (2012) model a dynamic theme park navigation problem as the SOP. Information such as current queuing times at various attractions and ride status affect the itinerary. A real-world theme park data set from Singapore is considered. A comprehensive study is done by considering non-peak and peak days.

Gunawan et al. (2014) focus on the problem of providing automatic tour guidance to a large leisure facility. This problem is treated as a variant of the TDOP. The travel time between two nodes depends on the time when the trip starts. Two real case studies from two theme parks in Asia are solved by the proposed algorithm.

Varakantham et al. (2015) address the problem of crowd congestion at certain venues like theme parks, museums and world expos as a variant of the Multi-agent OP. Multiple users are assumed to be selfish and move through the venue simultaneously considering each other's plans. This problem is modeled as a Selfish OP. The study is again applied to a theme park in Singapore.

Tricoire et al. (2010) present the MuPOPTW for the individual route planning of field workers and sales representatives (e.g. the pharmaceutic industry). Sales representatives have to visit their mandatory customers and they also have to consider some optional ones into their paths. Zhang et al. (2014) also model the pharmaceutical sales representative planning problem as the SOPTW. Pharmaceutical sales must decide which doctors to visit and the order to visit them in order to inform them about their products and encourage them to become an active prescriber.

Pietz and Royset (2013) formulate the Smuggler Search Problem (SSP) as an important special case of the GOP-RDR. The SSP is a path-constrained optimal search problem that deals with the decision of routing search vehicles through subsets of the area of interest in the presence of uncertain information about target whereabouts.

Van der Merwe et al. (2014) present the application of the COPTW in managing the allocation of resources during wildfires. During large wildfires, we have to make optimal use of limited resources (e.g. fire trucks). They apply the COPTW to a wildfire asset protection scenario in South Hobart, Tasmania (Australia).

The practical problem related to the application of integration of vehicle routing, inventory management and customer selection is studied by Vansteenwegen and Mateo (2014). The problem is called the Single-Vehicle Cyclic Inventory Routing Problem (SV-CIRP). The objective is to minimize the distribution and inventory costs at the customers and to maximize the collected scores, by selecting a subset of customers and determining the quantity to deliver to each customer and the vehicle paths, while avoiding stock-outs. This problem can be considered as the Inventory OP (IOP).

Other than those mentioned above, we also observe that several papers discuss about the possible applications of the OP and variants without further applying their proposed algorithms in solving the practical problems. Some of them are applied to modified benchmark instances. For example, Campbell et al. (2011), Papapanagiotou et al. (2014), Tarantilis et al. (2013), Archetti et al. (2014b) and Archetti et al. (2015) briefly describe a practical application in logistics.

4.4. Discussion

Two main types of applications of the OP summarized in Section 4 are the mobile crowdsourcing problem and the TTDP. Both applications have similar characteristics where users/individuals would like to find the best paths for maximiz-

Table 15
Papers on other applications.

Problem	Reference	Algorithm	Related OP
Theme park navigation problem	Lau et al. (2012)	Hybrid Variable Neighborhood Search and Simulated Annealing	DSOP
	Gunawan et al. (2014)	Iterated Local Search	TDOP
	Varakantham et al. (2015)	Mixed Integer Linear Programming - Sample Average Approximation	SeOP
Sales representative planning problem	Tricoire et al. (2010)	Variable Neighborhood Search	MuPOPTW
	Zhang et al. (2014)	Variable Neighborhood Search	SOPTW
Smuggler search problem	Pietz and Royset (2013)	Hybrid Branch-and-bound and SSP heuristic	GOP-RDR
Wildfire routing problem	Van der Merwe et al. (2014)	Exact algorithm	COPTW
Integration of vehicle routing, inventory management and customer selection problems	Vansteenwegen and Mateo (2014)	Iterated Local Search	IOP

ing their benefits. We found out that several mobile and web-based decision support applications have been introduced, such as ROSE (Ludwig et al., 2009), City Trip Planner (http://www.citytripplanner.com) (Vansteenwegen et al., 2011b) and eCOMPASS (http://ecompass.aegean.gr/) (Gavalas et al., 2015a). Other applications that have attracted some interest are in the MICE (Meetings, Incentives, Conferences, and Exhibitions) industry, amusement parks and integrated logistic problems.

For logistic problems, the mostly used basic model remains the vehicle routing problem. Nevertheless, the OP allows to include an extra layer in decision support, about which customers to select (instead of assuming that this has been decided beforehand). Moreover, the variety in applications discussed in this section illustrates that this extra layer of selecting ‘customers’ becomes more and more important in order to model decision problems from practice.

5. Conclusion

In this paper, we focus on the most recent papers about the OP and its variants. During the last decade, several extensions and variants of the OP have been introduced and were not really surveyed yet. Those cover the Time Dependent OP, the Generalized OP, the Stochastic OP and others. Some practical applications have also been modeled as an OP or its variants. We summarize several interesting applications which are related to the mobile crowdsourcing problem, the Tourist Trip Design Problem, the theme park navigation problem and others. A number of tables are introduces in order to present brief overviews of the recently published papers about variants of the OP.

In most cases, the OP is considered as a single agent problem where the problem is solved in a centralized manner. Recently, the problem is extended into a multi-agent level, namely the Multi-agent OP. Each agent is treated as a self-interested agent who only focuses on maximizing his/her respective scores. The main challenge for future work is to seek an equilibrium solution rather than a centralized optimal solution. This multi-agent point of view will be useful in modeling real-world applications in the MICE (Meetings, Incentives, Conferences, and Exhibitions) industry, amusement parks and museums. In the context of routing problems, this corresponds to multiple vehicles that compete or cooperate with each other.

Other future research on time-dependent travel times, multi-constraints and multi-objectives of the OP variants, in order to capture more realistic scenarios, would also be interesting. Future research efforts should be devoted to the development of appropriate solution techniques for these challenging variants. The recent trend of hybridizing exact algorithms and metaheuristics seems very promising in this matter.

Finally, developing web and mobile client application provides considerable room for research since many researchers have moved towards the applications of the OP in real world problems, such as the crowdsourcing problem and the TTDP. Moreover, researchers should be encouraged to make their solvers for the (T)OP(TW) available online, so other researchers can use these solvers and focus on further developments and/or other extensions instead of having to implement their own solver for the basic problem. Until now, unfortunately, only one solver, for the TDOP, is freely available.

Acknowledgments

This research is supported by the National Research Foundation, Prime Minister’s Office, Singapore under its International Research Centres in Singapore Funding Initiative.

References

Abbaspour, R. A., & Samadzadegan, F. (2011). Time-dependent personal tour planning and scheduling in metropolises. *Expert Systems with Applications, 38*(10), 12439–12452.
Afsar, H. M., & Labadie, N. (2013). Team orienteering problem with decreasing profits. *Electronic Notes in Discrete Mathematics, 41*, 285–293.
Angelelli, E., Archetti, C., & Vindigni, M. (2014). The clustered orienteering problem. *European Journal of Operational Research, 238*(2), 404–414.
Archetti, C., Bianchessi, N., & Speranza, M. G. (2013a). The capacitated team orienteering problem with incomplete service. *Optimization Letters, 7*(7), 1405–1417.
Archetti, C., Bianchessi, N., & Speranza, M. G. (2013b). Optimal solutions for routing problems with profits. *Discrete Applied Mathematics, 161*(4–5), 547–557.
Archetti, C., Bianchessi, N., & Speranza, M. G. (2014a). The split delivery capacitated team orienteering problem. *Networks, 63*(1), 16–33.
Archetti, C., Corberán, Á., Plana, I., Sanchis, J. M., & Speranza, M. G. (2015). A matheuristic for the team orienteering arc routing problem. *European Journal of Operational Research, 245*(2), 392–401.
Archetti, C., Corberán, Á., Plana, I., Sanchis, J. M., & Speranza, M. G. (2016). A branch-and-cut algorithm for the orienteering arc routing problem. *Computers & Operations Research, 66*, 95–104.
Archetti, C., Feillet, D., Hertz, A., & Speranza, M. G. (2009). The capacitated team orienteering and profitable tour problems. *Journal of the Operational Research Society, 60*, 831–842.
Archetti, C., Hertz, A., & Speranza, M. G. (2006). A tabu search algorithm for the split delivery vehicle routing problem. *Transportation Science, 40*(1), 64–73.
Archetti, C., & Speranza, M. G. (2014). Arc routing problems with profits. In Á. Corberán, & G. Laporte (Eds.), *Arc routing: Problems, methods, and applications*. In *MOS-SIAM Series on Optimization, Ch. 12* (pp. 281–300). Philadelphia: SIAM.
Archetti, C., Speranza, M. G., Corberán, Á., Sanchis, J. M., & Plana, I. (2014b). The team orienteering arc routing problem. *Transportation Science, 48*(3), 442–457.
Archetti, C., Speranza, M. G., & Vigo, D. (2014c). Vehicle routing problems with profits. In P. Toth, & D. Vigo (Eds.), *Vehicle routing: Problems, methods, and applications*. In *MOS-SIAM Series on Optimization, Ch. 10* (pp. 273–298). Philadelphia: SIAM.
Bouly, H., Dang, D.-C., & Moukrim, A. (2010). A memetic algorithm for the team orienteering problem. *4OR, 8*(1), 49–70.
Boussier, S., Feillet, D., & Gendreau, M. (2007). An exact algorithm for the team orienteering problem. *4OR, 5*(3), 211–230.
Campbell, A. M., Gendreau, M., & Thomas, B. W. (2011). The orienteering problem with stochastic travel and service times. *Annals of Operations Research, 186*(1), 61–81.
Campos, V., Martí, R., Sánchez-Oro, J., & Duarte, A. (2014). GRASP with path relinking for the orienteering problem. *Journal of the Operational Research Society, 65*(12), 1800–1813.

Chao, I.-M., Golden, B. L., & Wasil, E. A. (1996a). A fast and effective heuristic for the orienteering problem. *European Journal of Operational Research, 88*(3), 475–489.

Chao, I.-M., Golden, B. L., & Wasil, E. A. (1996b). The team orienteering problem. *European Journal of Operational Research, 88*(3), 464–474.

Chekuri, C., Korula, N., & Pál, M. (2012). Improved algorithms for orienteering and related problems. *ACM Transactions on Algorithms, 8*, 661–670.

Chen, C., Cheng, S.-F., Gunawan, A., Misra, A., Dasgupta, K., & Chander, D. (2014a). TRACCS: Trajectory-aware coordinated urban crowd-sourcing. In *Proceedings of the 2nd AAAI conference on human computation and crowdsourcing (HCOMP 2014), Pittsburgh, USA*.

Chen, C., Cheng, S.-F., & Lau, H. C. (2014b). Multi-agent orienteering problem with time-dependent capacity constraints. *Web Intelligence and Agent Systems, 12*(4), 347–358.

Chen, C., Cheng, S.-F., Lau, H. C., & Misra, A. (2015). Towards city-scale mobile crowdsourcing: Task recommendations under trajectory uncertainties. In *Proceedings of the international joint conference on artificial intelligence (ijcai-2015), Buenos Aires, Argentina*.

Cura, T. (2014). An artificial bee colony algorithm approach for the team orienteering problem with time windows. *Computers & Industrial Engineering, 74*, 270–290.

Dang, D.-C., El-Hajj, R., & Moukrim, A. (2013a). A branch-and-cut algorithm for solving the team orienteering problem. In C. Gomes, & M. Sellmann (Eds.), *Integration of AI and OR techniques in constraint programming for combinatorial optimization problems. Lecture Notes in Computer Science: 7874* (pp. 332–339). Springer.

Dang, D.-C., Guibadj, R. N., & Moukrim, A. (2011). A PSO-based memetic algorithm for the team orienteering problem. In C. Di Chio et al. (Ed.), *Applications of evolutionary computation*. In *Lecture Notes in Computer Science: 6625* (pp. 471–480). Springer.

Dang, D.-C., Guibadj, R. N., & Moukrim, A. (2013b). An effective PSO-inspired algorithm for the team orienteering problem. *European Journal of Operational Research, 229*(2), 332–344.

Divsalar, A., Vansteenwegen, P., & Cattrysse, D. (2013). The orienteering problem with hotel selection: a variable neighborhood search method. *International Journal of Production Economics, 145*(1), 150–160.

Divsalar, A., Vansteenwegen, P., Chitsaz, M., Sörensen, K., & Cattrysse, D. (2014a). Personalized multi-day trips to touristic regions: A hybrid GA-VND approach. In C. Blum, & G. Ochoa (Eds.), *Evolutionary computation in combinatorial optimisation*. In *Lecture Notes in Computer Science: vol. 8600* (pp. 194–205). Springer.

Divsalar, A., Vansteenwegen, P., Sorensen, K., & Cattrysse, D. (2014b). A memetic algorithm for the orienteering problem with hotel selection. *European Journal of Operational Research, 237*(1), 29–49.

Duque, D., Lozano, L., & Medaglia, A. L. (2015). Solving the orienteering problem with time windows via the pulse framework. *Computers & Operations Research, 54*, 168–176.

Erdoğan, G., Cordeau, J.-F., & Laporte, G. (2010). The attractive traveling salesman problem. *European Journal of Operational Research, 203*(1), 59–69.

Erdoğan, G., & Laporte, G. (2013). The orienteering problem with variable profits. *Networks, 61*(2), 104–116.

Evers, L., Glorie, K., van der Ster, S., Barros, A. I., & Monsuur, H. (2014). A two-stage approach to the orienteering problem with stochastic weights. *Computers & Operations Research, 43*, 248–260.

Feillet, D., Dejax, P., & Gendreau, M. (2005). Traveling salesman problems with profits. *Transportation Science, 39*(2), 188–205.

Ferreira, J., Quintas, A., & Oliveira, J. A. (2014). Solving the team orienteering problem: developing a solution tool using a genetic algorithm approach. In P. Krömer, M. Köppen, & G. Schaefer (Eds.), *Soft computing in industrial applications. Advances in Intelligent Systems and Computing: 223* (pp. 365–375). Springer.

Fischetti, M., Salazar-González, J. J., & Toth, P. (1998). Solving the orienteering problem through branch-and-cut. *INFORMS Journal on Computing, 10*, 133–148.

Fomin, F. V., & Lingas, A. (2002). Approximation algorithms for time-dependent orienteering. *Information Processing Letters, 83*, 57–62.

Gambardella, L. M., Montemanni, R., & Weyland, D. (2012). Coupling ant colony systems with strong local searches. *European Journal of Operational Research, 220*(3), 831–843.

Garcia, A., Arbelaitz, O., Vansteenwegen, P., Souffriau, W., & Linaza, M. T. (2010). Hybrid approach for the public transportation time dependent orienteering problem with time windows. In E. Corchado, M. G. Romay, & A. M. Savio (Eds.), *Hybrid artificial intelligence systems, lecture notes in artificial intelligence: vol. 6077* (pp. 151–158). Springer, Berlin, Germany.

Garcia, A., Linaza, M. T., Arbelaitz, O., & Vansteenwegen, P. (2009). Intelligent routing system for a personalised electronic tourist guide. In W. Höpken, U. Gretzel, & R. Law (Eds.), *Information and communication technologies in tourism 2009* (pp. 185–197). Springer.

Garcia, A., Vansteenwegen, P., Arbelaitz, O., Souffriau, W., & Linaza, M. T. (2013). Integrating public transportation in personalised electronic tourist guides. *Computers & Operations Research, 40*(3), 758–774.

Gavalas, D., Kasapakis, V., Konstantopoulos, C., Pantziou, G., Vathis, N., & Zaroliagis, C. (2015a). The eCOMPASS multimodal tourist tour planner. *Expert Systems with Applications, 42*(21), 7303–7316.

Gavalas, D., Konstantopoulos, C., Mastakas, K., & Pantziou, G. (2014a). A survey on algorithmic approaches for solving tourist trip design problems. *Journal of Heuristics, 20*(3), 291–328.

Gavalas, D., Konstantopoulos, C., Mastakas, K., Pantziou, G., & Tasoulas, Y. (2013). Cluster-based heuristics for the team orienteering problem with time windows. In V. Bonifaci, C. Demetrescu, & A. Marchetti-Spaccamela (Eds.), *Experimental algorihtms. Lecture Notes in Computer Science: 7933* (pp. 390–401). Springer.

Gavalas, D., Konstantopoulos, C., Mastakas, K., Pantziou, G., & Vathis, N. (2014b). Efficient heuristics for the time dependent team orienteering problem with time windows. In P. Gupta, & C. Zaroliagis (Eds.), *Applied algorithms*. In *Lecture Notes in Computer Science: vol. 8321* (pp. 152–163). Springer.

Gavalas, D., Konstantopoulos, C., Mastakas, K., Pantziou, G., & Vathis, N. (2015b). Approximation algorithms for the arc orienteering problem. *Information Processing Letters, 115*(2), 313–315.

Geem, Z. W., Kim, J. H., & Loganathan, G. V. (2001). A new heuristic optimization algorithm: harmony search. *Simulation, 76*(2), 60–68.

Geem, Z. W., Tseng, C. L., & Park, Y. (2005). Harmony search for generalized orienteering problem: best touring in china. In L. Wang, K. Chen, & Y. S. Ong (Eds.), *Advances in natural computation. Lecture Notes in Computer Science: 3612* (pp. 741–750). Springer.

Goldberg, D. E. (1989). *Genetic algorithms in search, optimization, and machine learning*. Massachusetts: Addison-Wesley, Reading.

Golden, B. L., Levy, L., & Vohra, R. (1987). The orienteering problem. *Naval Research Logistics, 34*(3), 307–318.

Golden, B. L., Wang, Q., & Liu, L. (1988). A multifaceted heuristic for the orienteering problem. *Naval Research Logistics, 35*(3), 359–366.

Gunawan, A., Lau, H. C., & Lu, K. (2015a). An iterated local search algorithm for solving the orienteering problem with time windows. In G. Ochoa, & F. Chicano (Eds.), *Evolutionary computation in combinatorial optimization. Lecture Notes in Computer Science: vol. 9026* (pp. 61–73). Springer.

Gunawan, A., Lau, H. C., & Lu, K. (2015b). SAILS: hybrid algorithm for the team orienteering problem with time windows. In *Proceedings of the 7th multidisciplinary international scheduling conference (MISTA 2015), Prague, Czech Republic* (pp. 276–295).

Gunawan, A., Lau, H. C., & Lu, K. (2015c). Well-tuned ILS for extended team orienteering problem with time windows. *LARC Technical Report Series*. Singapore Management University.

Gunawan, A., Yuan, Z., & Lau, H. C. (2014). A mathematical model and metaheuristics for time dependent orienteering problem. In *Proceedings of the 10th international conference of the practice and theory of automated timetabling (patat 2014), York, United Kingdom* (pp. 202–217).

Herzog, D., & Wörndl, W. (2014). A travel recommender system for combining multiple travel regions to a composite trip. In *Proceedings of the new trends in content-based recommender systems (CBRECSYS) workshop, 8th association for computing machinery ACM conference on recommender systems, California, USA*.

Howe, J. (2008). *Crowdsourcing: Why the power of the crowd is driving the future of business*. Crown Business.

Hu, Q., & Lim, A. (2014). An iterative three-component heuristic for the team orienteering problem with time windows. *European Journal of Operational Research, 232*(2), 276–286.

Ilhan, T., Iravani, S. M. R., & Daskin, M. S. (2008). The orienteering problem with stochastic profits. *IIE Transactions, 40*(4), 406–421.

Kantor, M. G., & Rosenwein, M. B. (1992). The orienteering problem with time windows. *Journal of the Operational Research Society, 43*(6), 629–635.

Ke, L., Zhai, L., Li, J., & Chan, F. T. S. (2015). Pareto mimic algorithm: an approach to the team orienteering problem. *Omega, 61*, 155–166.

Keshtkaran, M., Ziarati, K., Bettinelli, A., & Vigo, D. (2016). Enhanced exact solution methods for the team orienteering problem. *International Journal of Production Research, 54*(2), 591–601.

Kulturel-Konak, S., Norman, B. A., Coit, D. W., & Smith, A. E. (2004). Exploiting tabu-search memory in constrained problems. *INFORMS Journal on Computing, 16*(3), 241–254.

Labadie, N., Mansini, R., Melechovskỳ, J., & Wolfler Calvo, R. (2011). Hybridized evolutionary local search algorithm for the team orienteering problem with time windows. *Journal of Heuristics, 17*(6), 729–753.

Labadie, N., Mansini, R., Melechovskỳ, J., & Wolfler Calvo, R. (2012). The team orienteering problem with time windows: an LP-based granular variable neighborhood search. *European Journal of Operational Research, 220*(1), 15–27.

Lambert III, T. J., Epelman, M. A., & Smith, R. L. (2005). A fictitious play approach to large-scale optimization. *Operations Research, 53*(3), 477–489.

Laporte, G., & Rodríguez-Martín, I. (2007). Locating a cycle in a transportation or a telecommunications network. *Networks, 50*(1), 92–108.

Lau, H. C., Yeoh, W., Varakantham, P., Nguyen, D. T., & Chen, H. (2012). Dynamic stochastic orienteering problems for risk-aware applications. In *Proceedings of the 28th conference annual conference on uncertainty in artificial intelligence (UAI-12)* (pp. 448–458). Corvallis, Oregon

Li, J. (2012). Research on team orienteering problem with dynamic travel times. *Journal of Software, 7*(2), 249–255.

Liang, Y. C., Kulturel-Konak, S., & Lo, M. H. (2013). a multiple-level variable neighborhood search approach to the orienteering problem. *Journal of Industrial and Production Engineering, 30*(4), 238–247.

Liao, C. C., & Hsu, C. H. (2013). A detour planning algorithm in crowdsourcing systems for multimedia content gathering. In *Proceedings of the 5th workshop on mobile video (MOVID'13), Oslo, Norway* (pp. 55–60).

Lin, S. W. (2013). Solving the team orienteering problem using effective multi-start simulated annealing. *Applied Soft Computing, 13*(2), 1064–1073.

Lin, S. W., & Yu, V. F. (2012). A simulated annealing heuristic for the team orienteering problem with time windows. *European Journal of Operational Research, 217*(1), 94–107.

Lin, S. W., & Yu, V. F. (2015). A simulated annealing heuristic for the multiconstraint team orienteering problem with multiple time windows. *Applied Soft Computing, 37*, 632–642.
Lozano, L., Duque, D., & Medaglia, A. L. (2015). An exact algorithm for the elementary shortest path problem with resource constraints. *Transportation Science, 50*(1), 348–357.
Lu, Y., & Shahabi, C. (2015). An arc orienteering algorithm to find the most scenic path on a large-scale road network. In *Proceedings of the 23rd ACM SIGSPATIAL international conference on advances in geographic information systems, Washington, USA*.
Ludwig, B., Zenker, B., & Schrader, J. (2009). Recommendation of personalized routes with public transport connections. In D. Tavangarian, T. Kirste, D. Timmermann, U. Lucke, & D. Versick (Eds.), *Intelligent interactive assistance and mobile multimedia computing. Communications in Computer and Information Science: vol. 53* (pp. 97–107). Springer.
Malucelli, F., Giovannini, A., & Nonato, M. (2015). Designing single origin-destination itineraries for several classes of cycle-tourists. *Transportation Research Procedia, 10*, 413–422.
Marinakis, Y., Politis, M., Marinaki, M., & Matsatsinis, N. (2015). A memetic-GRASP algorithm for the solution of the orienteering problem. In H. A. Le Thi, T. P. Dinh, & N. T. Nguyen (Eds.), *Modelling, computation and optimization in information systems and management sciences*. In *Advances in Intelligent Systems and Computing: 360* (pp. 105–116). Springer.
Miller, C. E., Tucker, A. W., & Zemlin, R. A. (1960). Integer programming formulations and travelling salesman problems. *Journal of the Association for Computing Machinery, 7*, 326–329.
Montemanni, R., & Gambardella, L. M. (2009). Ant colony system for team orienteering problem with time windows. *Foundations of Computing and Decision Sciences, 34*(4), 287–306.
Mousselly-Sergieh, H., Watzinger, D., Huber, B., Döller, M., Egyed-Zsigmond, E., & Kosch, H. (2014). World-wide scale geotagged image dataset for automatic image annotation and reverse geotagging. In *Proceedings of the 5th ACM multimedia systems conference (MMSys'14), Singapore* (pp. 47–52).
Muthuswamy, S., & Lam, S. S. (2011). Discrete particle swarm optimization for the team orienteering problem. *Memetic Computing, 3*(4), 287–303.
Papapanagiotou, V., Montemanni, R., & Gambardella, L. M. (2014). Objective function evaluation methods for the orienteering problem with stochastic travel and service times. *Journal of Applied Operations Research, 6*(1), 16–29.
Pessoa, A., Poggi, M., & Uchoa, E. (2009). A robust branch-cut-and-price algorithm for the heterogeneous fleet vehicle routing problem. *Networks, 54*(4), 167–177.
Pietz, J., & Royset, J. O. (2013). Generalized orienteering problem with resource dependent rewards. *Naval Research Logistics, 60*(4), 294–312.
Poggi, M., Viana, H., & Uchoa, E. (2010). The team orienteering problem: Formulations and branch-cut and price. In *Proceedings of the 10th workshop on algorithmic approaches for transportation modelling, optimization, and systems (atmos'10), Liverpool, United Kingdom* (pp. 142–155).
Punnen, A. P. (2007). The traveling salesman problem: applications, formulations and variations. In G. Gutin, & A. P. Punnen (Eds.), *The traveling salesman problem and its variations. Combinatorial Optimization: 12* (pp. 1–28). Springer.
Ramesh, R., & Brown, K. M. (1991). An efficient four-phase heuristic for the generalized orienteering problem. *Computers & Operations Research, 18*(2), 151–165.
Ramirez-Marquez, J. E., Kulturel-Konak, S., & Sanseverino, C. M. R. (2010). Probabilistic solution discovery algorithm for the orienteering problem. *International Journal of Industrial and Systems Engineering, 6*(1), 45–61.
Rego, C., & Glover, F. (2010). Ejection chain and filter-and-fan methods in combinatorial optimization. *Annals of Operations Research, 175*(1), 77–105.
Reinelt, G. (1991). TSPLIB - a traveling salesman problem library. *ORSA Journal on Computing, 3*(4), 376–384.
Resende, M. G. C., & Ribeiro, C. C. (2003). Greedy randomized adaptive search procedure. In F. Glover, & G. Kochenberger (Eds.), *State-of-the-art handbook in metaheuristics* (pp. 219–250). Kluwer Academic Publishers.
Righini, G., & Salani, M. (2009). Decremental state space relaxation strategies and initialization heuristics for solving the orienteering problem with time windows with dynamic programming. *Computers & Operations Research, 36*(4), 1191–1203.
Salazar-Aguilar, M. A., Langevin, A., & Laporte, G. (2014). The multi-district team orienteering problem. *Computers & Operations Research, 41*, 76–82.
Schilde, M., Doerner, K. F., Hartl, R. F., & Kiechle, G. (2009). Metaheuristics for the biobjective orienteering problem. *Swarm Intelligence, 3*(3), 179–201.
Sevkli, Z., & Sevilgen, F. E. (2010a). Discrete particle swarm optimization for the orienteering problem. In *Proceedings of the IEEE congress on evolutionary computation (CEC 2010), Barcelona, Spain* (pp. 3234–3241).
Sevkli, Z., & Sevilgen, F. E. (2010b). StPSO: Strengthened particle swarm optimization. *Turkish Journal of Electrical Engineering & Computer Sciences, 18*(6), 1095–1114.
Silberholz, J., & Golden, B. L. (2010). The effective application of a new approach to the generalized orienteering problem. *Journal of Heuristics, 16*(3), 393–415.
Solomon, M. M. (1987). Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research, 35*(2), 254–265.
Souffriau, W., Vansteenwegen, P., Vanden Berghe, G., & Van Oudheusden, D. (2011). The planning of cycle trips in the province of East Flanders. *Omega, 39*(2), 209–213.
Souffriau, W., Vansteenwegen, P., Vanden Berghe, G., & Van Oudheusden, D. (2013). The multiconstraint team orienteering problem with multiple time windows. *Transportation Science, 47*(1), 53–63.
Souffriau, W., Vansteenwegen, P., Vertommen, J., Vanden Berghe, G., & Van Oudheusden, D. (2008). A personalised tourist trip design algorithm for mobile tourist guides. *Applied Artificial Intelligence, 22*(10), 964–985.
Sylejmani, K., & Dika, A. (2011). Solving touristic trip planning problem by using taboo search approach. *International Journal of Computer Science, 8*(5), 139–149.
Sylejmani, K., Muhaxhiri, A., Dika, A., & Ahmedi, L. (2014). Solving tourist trip planning problem via a simulated annealing algorithm. In P. Biljanović, Z. Butković, K. Skala, & S. Golubić (Eds.), *Proceedings of the 37th international convention on information and communication technology electronics and microelectronics (mipro 2014), Opatija, Croatia* (pp. 1124–1129).
Tarantilis, C. D., Stavropoulou, F., & Repoussis, P. P. (2013). The capacitated team orienteering problem: a bi-level filter-and-fan method. *European Journal of Operational Research, 224*(1), 65–78.
Tricoire, F., Romauch, M., Doerner, K. F., & Hartl, R. F. (2010). Heuristics for the multi-period orienteering problem with multiple time windows. *Computers & Operations Research, 37*(2), 351–367.
Tsiligirides, T. (1984). Heuristic methods applied to orienteering. *Journal of the Operational Research Society, 35*(9), 797–809.
Ulusoy, G. (1985). The fleet size and mixed problem for capacitated arc routing. *European Journal of Operational Research, 22*(3), 329–337.
Van der Merwe, M., Minas, J. P., Ozlen, M., & Hearne, J. W. (2014). The cooperative orienteering problem with time windows. *Optimization Online*.
Vansteenwegen, P., & Mateo, M. (2014). An iterated local search algorithm for single-vehicle cyclic inventory. *European Journal of Operational Research, 237*(3), 802–813.
Vansteenwegen, P., Souffriau, W., & Van Oudheusden, D. (2011a). The orienteering problem: A survey. *European Journal of Operational Research, 209*(1), 1–10.
Vansteenwegen, P., Souffriau, W., Vanden Berghe, G., & Van Oudheusden, D. (2009). Iterated local search for the team orienteering problem with time windows. *Computers & Operations Research, 36*(12), 3281–3290.
Vansteenwegen, P., Souffriau, W., Vanden Berghe, G., & Van Oudheusden, D. (2011b). The city planner: An expert system for tourists. *Expert Systems with Applications, 38*(6), 6540–6546.
Vansteenwegen, P., & Van Oudheusden, D. (2007). The mobile tourist guide: an OR opportunity. *OR Insights, 20*(3), 21–27.
Varakantham, P., & Kumar, A. (2013). Optimization approaches for solving chance constrained stochastic orienteering problems. In P. Perny, M. Pirlot, & A. Tsoukiàs (Eds.), *Algorithmic decision theory*. In *Lecture Notes in Computer Science: vol. 8176* (pp. 387–398). Springer.
Varakantham, P., Mostafa, H., Fu, N., & Lau, H. C. (2015). DIRECT: A scalable approach for route guidance in selfish orienteering problems. In *Proceedings of the 14th international conference on autonomous agents and multiagent systems (AAMAS 2015), Istanbul, Turkey*.
Verbeeck, C., Sörensen, K., Aghezzaf, E. H., & Vansteenwegen, P. (2014a). A fast solution method for the time-dependent orienteering problem. *European Journal of Operational Research, 236*(2), 419–432.
Verbeeck, C., Vansteenwegen, P., & Aghezzaf, E. H. (2014b). An extension of the arc orienteering problem and its application to cycle trip planning. *Transportation Research Part E, 68*, 64–78.
Wang, Q., Sun, X., & Golden, B. L. (1996). Using artificial neural networks to solve generalized orienteering problems. In C. Dagli, M. Akay, C. Chen, & B. Fernández (Eds.), *Intelligent engineering systems through artificial neural networks: vol. 6* (pp. 1063–1068). ASME Press, New York.
Wang, X., Golden, B. L., & Gulczynski, D. (2014). A worst-case analysis for the split delivery capacitated team orienteering problem with minimum delivery amounts. *Optimization Letters, 8*(8), 2349–2356.
Wang, X., Golden, B. L., & Wasil, E. A. (2008). Using a genetic algorithm to solve the generalized orienteering problem. In B. L. Golden, S. Raghavan, & E. A. Wasil (Eds.), *The vehicle routing problem: Latest advances and new challenges*. In *Operations Research/Computer Science Interfaces: vol. 43* (pp. 263–274). Springer.
Yu, J., Aslam, J., Karaman, S., & Rus, D. (2015). Anytime planning of optimal schedules for a mobile sensing robot. In *Proceedings of the IEEE/RSJ international conference on intelligent robots and systems (IROS 2015)* (pp. 5279–5286). Hamburg, Germany.
Yu, J., Schwager, M., & Rus, D. (2014). Correlated orienteering problem and its application to informative path planning for persistent monitoring tasks. In *Proceedings of the IEEE/RSJ international conference on intelligent robots and systems (IROS 2014)* (pp. 342–349). Chicago, Illinois.
Yuen, M., King, I., & Leung, K. (2011). A survey of crowdsourcing systems. In *Proceedings of the IEEE international conference on social computing (Socialcom'11), Boston, Massachusetts* (pp. 766–773).
Zhang, S., Ohlmann, J. W., & Thomas, B. W. (2014). A priori orienteering with time windows and stochastic wait times at customers. *European Journal of Operational Research, 239*(1), 70–79.