The Bi-objective Minimum Latency Problem with Profit Collection and Uncertain Travel Times

Maria Elena Bruni[1], Sara Khodaparasti[1] and Samuel Nucamendi-Guillén[2]
[1]*Department of Mechanical, Energy and Management Engineering, Unical, Italy*
[2]*Universidad Panamericana, Facultad de Ingeniería, Álvaro del Portillo 49, Zapopan, Jalisco, 45010, Mexico*
{*mariaelena.bruni, sara.khodaparasti*}*@unical.it, snucamendi@up.edu.mx*

Keywords: Minimum Latency Problem, Profit, Bi-objective Optimization, Uncertainty, Risk.

Abstract: This paper introduces a new bi-objective minimum latency problem with profit collection, where routes must be constructed in order to maximize the collected profit and to minimize the total latency. These objectives are usually conflicting. Thus, considering some important features, as the segmentation of the customers into two classes, mandatory and optional, and the presence of uncertain travel times, we follow a bi-objective approach, aiming to compute a set of Pareto-optimal alternatives with different trade-offs for a decision-maker to choose from. In order to address this computationally challenging problem, we propose a Multi-Objective Iterated Local Search. Computational results confirm the practicality of the algorithm, in terms of the quality of the solutions, and its computational efficiency in terms of time spent. We conclude that the algorithm finds good-quality solutions for small and medium-size instances.

1 INTRODUCTION

This paper models and solves a new routing problem of practical importance, which considers customers with different service level agreements. The frequent customers are mandatory to be serviced, whilst the service requests of the non-frequent customers might be either rejected or accepted. The attractiveness of these additional customers relies on the potential additional profit that can be gained. A generic application of the problem we are considering is the design of routes for technicians for repair and maintenance operations. Mandatory customers are requiring preventive maintenance operations, whereas optional customers are requiring a repair service. In this application, vehicles are used only for carrying material and personnel. Thus, we can suppose that the vehicle capacity is unlimited. There are other applicative contexts in which the company has to regularly visit customers with long-term relations, whereas potential customers, usually located close to the existing ones, can be serviced, in an effort to expand the existing customer base. A similar setting is also faced by small package shipping companies, where commercial customers need to be regularly visited, while residential customers are only visited on an ad hoc basis. The aforementioned problems pose the same challenge: designing a set of routes with the aim of visiting all the mandatory customers and, at the same time, determining the subset of the potential customers that will be included in the routing plans. Although such problems are frequently used to model real cases, they are often modelled as single objective models, despite the fact that in the majority of applications are multi-objective in nature. Two conflicting objectives can be considered relevant in our case. The first is to maximize the total collected profit, while the second is to minimize the total arrival time to the customers. Combining these conflicting objectives into one single objective is questionable, since they are expressed in different measurement units, motivating the modelling of the problem as a bi-objective one.

The main contributions of this paper are:

- The introduction of the bi-objective minimum latency problem with profit collection, considering realistic features such as the presence of optional customers and stochastic travel times.
- While modeling this problem, we consider a risk-averse measure for the total arrival time, instead of the widely used expected risk measure.
- We present a more general and risk-averse approach, which includes the well known Conditional Value at Risk (CVaR, for short) as a special case, providing a unified framework for dealing with risk.

- We design and implement an iterated greedy procedure to efficiently deal with instances of reasonable size that can be used for a broad class of risk measures.

The paper is organized as follows. The next section is dedicated to the literature review. Section 3 presents the problem description, formalized in the Appendix. Section 4 is devoted to the description of a solution approach. We discuss the effectiveness of the method and report computational results in Section 5. We conclude in Section 6.

2 LITERATURE REVIEW

The minimum latency problem (MLP) or travelling repairman problem (TRP) is one of the most famous customer-centric routing problems. It consists in finding a tour starting from a depot node, which minimizes the sum of the elapsed times (or latencies) to reach a given set of nodes. The problem arises in situations in which the arrival time has a crucial role in the customers satisfaction and it has recently attracted the attention of the researchers, due to its importance in applicative fields such as emergency logistics (Bruni et al., 2018b), delivery logistics (Bruni et al.,), and manufacturing contexts such as machine scheduling (Bruni et al., 2019).

This problem has been extensively studied by a large number of researchers who proposed several exact and non-exact approaches. Lucena (Lucena, 1990) and Bianco et al. (Bianco et al., 1993) proposed early exact enumerative algorithms, in which lower bounds are derived using a Lagrangian relaxation. Fischetti et al. (Fischetti et al., 1993) proposed an enumerative algorithm that makes use of lower bounds obtained from a linear integer programming formulation. Different mixed integer programming formulations with various families of valid inequalities have been proposed in the last years (Bigras et al., 2008; Ezzine et al., 2010; Méndez-Díaz et al., 2008; Van Eijl, 1995). Salehipour et al. (Salehipour et al., 2011) first proposed a simple composite algorithm based on a GRASP, improved with a variable neighborhood search procedure. In (Mladenović et al., 2013a), Mladenović et al. presented a general variable neighborhood search metaheuristic enhanced with a move evaluation procedure facilitating the update of the incumbent solution. Silva et al. (Silva et al., 2012) presented a composite multi-start metaheuristic approach consisting of a GRASP and a randomized variable neighborhood descent algorithm. A direct extension of the TRP/MLP is the multiple traveling repairman problem (k-TRP) that considers identical vehicles. Although many researchers have studied the TRP, the literature on the multiple vehicle case is surprisingly limited. Recently, Nucamendi-Guillén et al. (Nucamendi-Guillén et al., 2016; Nucamendi-Guillén et al., 2018) presented an efficient new formulation, defined on a multi-level network, for the deterministic k-traveling repairman problem enhanced by an iterative greedy metaheuristic. Several metaheuristic algorithms have been designed for efficiently solving routing problems with cumulative costs and its variants (Mladenović et al., 2013b; Ngueveu et al., 2010; Ribeiro and Laporte, 2012; Rivera et al., 2015).

The k-Traveling Repairmen Problem with Profits (k-TRPP) has been introduced by Dewilde et al. (Dewilde et al., 2013) as an extension of the TRP where the service at each node is rewarded with a non-negative profit, which decreases with arrival time at the node. Recently, in (Yongliang et al., 2019) a population based hybrid evolutionary search algorithm has been proposed for solving the problem, combining a randomized greedy construction method for initial solution generation and a dedicated variable neighborhood search for local optimization. Although several contributions have addressed uncertainty in routing problems (Beraldi et al., 2005; Bruni et al., 2014; Beraldi et al., 2015a; Beraldi et al., 2015b) only a few contributions focused on incorporating uncertainty in the k-TRPP (Bruni et al., 2018a; Beraldi et al., 2019; Bruni et al., 2020). Moreover, all of the aforementioned works focused on the single-objective version of the problem, seeking for a trade-off between reward and variance, two different objectives that are not calculated with the same metric.

To the best of our knowledge, the only two contributions dealing with the multi-objective MLP are (Arellano-Arriaga et al., 2019; Arellano-Arriaga et al., 2017). Both papers consider a bi-objective approach for the MLP, considering a single-vehicle tour and minimising the travel time (as a measure of distance) and the latency of that tour. In this paper, we address the problem under a risk-averse perspective considering a fleet of vehicles, the profits and the presence of optional customers. To the best of out knowledge, the the problem studied in this paper has never been tackled before.

3 PROBLEM FORMULATION

Let consider an undirected graph $G = V, E$ where $V = \{0, 1, 2, \dots, n\}$ corresponds to the node set and E denotes the edge set. Node 0 denotes the depot and $V' = \{1, 2, \dots, n\}$ represents the set of customers further partitioned into two subsets: M is the set of mandatory customers, while O is the set of optional customers. For each demand node $i \in V'$, a profit p_i is defined. Additionally, there is a homogeneous fixed fleet of K uncapacitated vehicles, dispatched from the depot and that can serve any route assigned.

The aim is to design vehicle routes for serving a mix of regular and on the spot customers, while ensuring that the arrival time at the customers is minimized and the profit collected is maximized. To this end, this paper addresses a combined minimum latency and profit maximizing repairman problem through a bi-objective model that captures the profit collecting nature, as well as the main feature of the minimum latency problem.

In particular, the first objective function is the total profit collected. Let assume that we have a set of routes π^k, $k = 1, \ldots K$, then, the collected profit can be expressed as:

$$P = \sum_{k=1}^{K} \sum_{i \in \pi^k} p_i$$

Now, let assume that each edge $l \in E$ has an associated random travel time $\tilde{t}_l$, with a given mean μ_l and variance σ_l^2. When the travel times are considered random, the arrival time of each vehicle at generic node i is itself a random variable (denoted with $\tilde{t}_i$). In particular, the arrival time at each node is the sum of the travel times associated to the links $l \in \pi_i^k$ i.e. belonging to the subpath connecting the depot to the node i. The total arrival time, defined as

$$T = \sum_{k=1}^{K} \sum_{i \in \pi^k} \tilde{t}_i$$

is itself a random variable.

Since the decision-maker is risk-averse when making a decision, the problem does not merely entail the minimization of the expected arrival time, but it must also consider the decision maker's attitude against risk.

Formally, a risk measure is a map $\rho : \mathcal{X} -> \mathcal{R}$ that attaches a scalar value to each random variable $X : \Omega -> \mathcal{R}$, governed by a probability distribution function F_X, whose moment-generating function $M_X z = \mathbb{E} e^{zX}$ exists for all $z \geq 0$. Artzner et al. (Artzner et al., 1999) stated a set of properties that should be desirable for any risk measure. The four axioms they stated are: Monotonicity, Translation equivariance, Subadditivity and Positive Homogeneity. Given two random variables X and Y and a risk function, ρ, we can define the properties as follows.

- Monotonicity- A risk measure is monotone, if for all $X, Y : X \leq Y$ $\rho X \leq \rho Y$, i.e., higher losses mean higher risk
- Translation Equivariance- A risk measure is translation equivariant, if for all X, and scalars $c \in \mathbb{R}$: $\rho X + c = \rho X + c$, i.e., increasing (or decreasing) the loss increases (decreases) the risk by the same amount
- Subadditivity- A risk measure is subadditive, if for all X, Y $\rho X + Y \leq \rho X + \rho Y$, i.e., diversification decreases risk
- Positive Homogeneity- A risk measure is positively homogeneous, if for all X, $\lambda \geq 0$: $\rho \lambda X = \lambda \rho X$, i.e., doubling the size doubles the risk

Any risk measure which satisfies these axioms is said to be coherent (Artzner et al., 1999).

A general class of risk measures is represented by the spectral risk measures, first introduced by Acerbi (Acerbi, 2002). A spectral risk measure, denote by SRM_ϕ is a function parameterized by ϕ, a nondecreasing normalized right-continuous integrable probability density function, such that $\phi \geq 0$, and $\int_0^1 \phi p dp = 1$. The density function ϕ is also called an risk spectrum. It can be defined as follows:

$$SRM_\phi = \int_0^1 \phi p F^{-1} p dp = \int_0^1 \phi p VaR_p dp.$$

Spectral risk measures satisfy the properties of monotonicity, convexity, translation invariance and coherency. In most real-life applications, the probability distribution of the travel times is typically unknown and only indirectly observable through historical samples. A remedy for this difficulty is to adopt a distributionally robust approach, assuming that the probability distribution is merely known to belong to an ambiguity set, typically defined as the family $\mathbb{F}$ of all distributions that have known first and second moments. This ambiguity prompts us to investigate the quantification of the risk in this more general setting. In this case, solutions are evaluated under the worst-case over all the distributions in the family $\mathbb{F}$ and hence, consistent with the known moments. The resulting Worst-Case Spectral Risk Measure (WCSRM) represents a conservative (that is, pessimistic) approximation for the true (unknown) SRM. We can define the WCSRM as follows:

$$WCSRM = sup_{F \in \mathbb{F}} SRM_\phi.$$

As proposed in (Li, 2018), it can be proved that the WCSRM admits an elegant closed form expression:

$$WCSRM = \mu + \sigma \sqrt{\int_0^1 \phi^2 p dp - 1}$$

Considering the above definitions, the risk criterion reduces to the worst-case Conditional Value at

Risk[1], when

$$\phi p = \begin{cases} \frac{1}{1-\alpha} & \text{if } p > \alpha \\ 0 & \text{if } p \leq \alpha. \end{cases}$$

and we have ${}_0^1 \phi^2 p dp = \frac{1}{1-\alpha}$. In fact, we obtain the well known formula

$$WCVaR = sup_{F \in \mathbb{F}} CVaR_\alpha = \mu + \sigma \sqrt{\frac{\alpha}{1-\alpha}}.$$

A similar closed form (assuming Normal distributions) can also be obtained for another well-known risk measure, the Entropic VaR (EVaR), recently introduced in Ahmadi-Javid (Ahmadi-Javid, 2012a; Ahmadi-Javid, 2012b), which is the tightest possible upper bound for VaR and the CVaR. The EVaR of X with confidence level α is defined as follows:

$$EVaR_\alpha = inf_{z>0}\{z ln\big(M_X z^{-1} 1-\alpha\big)\} =$$

$$= inf_{z>0}\{z ln \mathbb{E}\big[\exp\big(\frac{X}{z}\big)\big] - z ln 1 - \alpha\}$$

Despite its apparent complexity, also the EVaR can be boiled down to the following closed form expression assuming normally distributed random variables:

$$EVaR_\alpha = \mu + \sigma\sqrt{-2ln1-\alpha}.$$

The above result provides a unified perspective on solving the problem under different risk measures with same objective function structure,

$$\mu + \Gamma\sigma$$

just by modifying the scale factor Γ of the standard deviation. Applying this risk measure to the total arrival time (T^{risk}) leads to the following objective function (for a fixed set of routes π^k, $k = 1, \ldots K$): The total completion time, defined as

$$T^{risk} = \sum_{k=1}^{K} \sum_{i \in \pi^k} \mathbb{E}\tilde{t}_i + \Gamma \sqrt{\sum_{k=1}^{K} \sum_{i \in \pi^k} \text{VAR}\tilde{t}_i}$$

where VAR represents the standard deviation of the total arrival time. The mathematical formulation of the problem is reported in the Appendix.

[1] Basically, CVaR is defined as the average of the $\alpha\%$ worst cases weighted with a uniform weight. More formally, the CVaR risk measure at a given confidence level $\alpha \in 0,1$, quantifies the expected loss of the random variable in the worst $1-\alpha\%$ of cases Hence:

$$CVaR_\alpha = \mathbb{E}X|X \geq VaR_\alpha.$$

If F_X is continuous, then we have

$$CVaR_\alpha = \frac{1}{1-\alpha} {}_\alpha^1 VaR_p dp$$

4 HEURISTIC PROCEDURE

Our heuristic approach for approximating the Pareto-front is based on three main procedures: a constructive phase, an improvement phase and a perturbation mechanism. The algorithm requires as input the following sets and parameters: the number of customers n, the number of vehicles K, the set of mandatory customers M and the set of optional customers O. In our algorithm, a solution is represented by $s = \{\pi^1, \pi^2, \ldots, \pi^k\}$. The pseudocode is shown in 1, where Sm denotes the set of mandatory nodes not yet visited, Sa the set of non-visited nodes and Sp the current partial solution. The algorithm ends when the maximum number of iterations (Maxiter) is reached.

Algorithm 1: Pseudo-code for the heuristic procedure.

```
Data: n, K
Initialization: s = {0}, iter := 0, MaxIter := T,
  Sa = V', Sm = M
ConstructiveProcedure
ImprovementProcedure
ParetoSetInsertion
while iter < MaxIter do
  | PerturbationProcedure
  | ConstructiveProcedure
  | ImprovementProcedure
  | ParetoSetInsertion
  | iter ← iter + 1
end
Filter the Pareto Front (F0)
Result: F0
```

In what follows, we will specialize each main step of the algorithm.

4.1 Constructive Procedure

This procedure is based on the parallel route building strategy originally proposed in (Potvin and Rousseau, 1993). The set of unrouted customers is denoted by (Sa), the current partial solution by (Sp) the cost matrix by C and the number of customers per route by n_l in Sp. The procedure also considers a generalized regret criterion for the selection of the customers to include in the solution.

For the first iteration, the constructive procedure starts with the empty routes in Sp. The procedure selects the customers in Sa with the greatest expected travel time with respect to the depot, giving priority to the mandatory customers. Once all of the routes have at least one customer, the procedure continues by sequentially inserting the remaining customers in Sa, always prioritizing the mandatory nodes. For this, the cost of insertion is computed based on the generalized regret measure described in (Nucamendi-Guillén et al., 2018) but considering the risk measure associated to the latency. The procedure ends when all of the mandatory customers have been assigned (independently of the remaining customers in *Sa*). For instance, if the first *Sm* customers assigned correspond to the mandatory ones, then the constructive procedure finalizes the assignment (even when there are *n* - *Sm* customers not assigned), and the solution created goes to the improvement procedure. On the other hand, optional nodes can be inserted before finishing the insertion of the mandatory nodes. Figure 2 shows the pseudocode for this procedure.

Algorithm 2: Outline of the Constructive Procedure.

Data: Sa, Sm
while $|Sm| > 0$ *or* $|Sa| > 0$ **do**
  if *there are empty routes* **then**
    Initialize them with customers $i \in Sa$ that have the highest values of $\pi_i - \mu_{0i}$
    $Sa := Sa \setminus i$ and if $i \in M$ $Sm := Sm \setminus i$
  end
  foreach *customer in Sa* **do**
    Determine the best insertion points over all the K partial routes
    Compute the regret between the values of all insertion points and the value of the best insertion point
  end
  Insert the first customer with the highest regret into its best place in **s**
  Update Sa, Sm
end
Compute the total collected profit *P and the total arrival time T* for the current solution *s*
return s

After the solution is improved, it is compared against the non-dominated solutions found so far. Details of the improvement phase are provided in the next section.

4.2 Improvement Procedure

After the initial construction is obtained, the solution is sent to a improvement procedure that applies five different local search strategies, arranged in two major groups: Intra_route neighborhoods (*Intra_RN*) and inter-route neighborhoods (*Inter_RN*). The neighborhoods used are:

- Swap move: operator that exchanges the position of two nodes, *i* and *j*, both belonging to the same route.
- Reallocation move: operator that removes a customer from its current position on the route and reinserts it in a different position on the same route.
- 2-opt move: Two adjacent edges are deleted in the tour, then the arcs are reversed and reconnected in a different way.
- Interchange move: Two nodes, each belonging to a different route, exchange their respective positions (when possible with respect to the remaining vehicles capacities).
- Insertion move: A customer is removed from its current position in the tour and inserted in a new position into a different route.

The *intra_RN* includes the swap, reallocation and 2-opt moves, whereas the *inter_RN* involves the interchange and insertion moves. The *intra_RN* starts by executing the swap move and it goes into a loop where the three local searches are iteratively executed, beginning with the reallocation move. This loop ends when none of the neighbourhoods can improve the current solution in at least one objective. On the other hand, the *inter_RN* performs first the interchange move and then the solution goes into a loop in which the Insertion and Interchange moves are applied iteratively. Similarly, the *inter_RN* procedure ends when none of the neighborhoods can improve their input solution. After finishing the improvement procedure, the solution is evaluated to verify if it can be candidate to be included in the Pareto Front. The procedure of evaluation determines if the solution is non-dominated, then it is inserted into the a list *CS*.

4.3 Perturbation Procedure

The perturbation procedure consists of a partial-removal mechanism that randomly selects a group of customers in *s* and assign them into *Sa* and the remaining clients in the routes are re-allocated to the first positions in their corresponding route preserving the order in which they were sequenced in the selected solution. In case the one or more customers removed belong to the mandatory set, then the indicator *Sm* is updated correspondingly.

4.4 Pareto Candidate Set Insertion

In every iteration, this mechanism evaluates if the solution obtained (after finalizing the improving procedure) is non-dominated with respect to the set of solutions found by the algorithm and stored in the candidate set (*CS*). It is evident that, for the first iteration, the mechanism immediately includes the solution obtained. From the second to the last iteration, the current solution is compared with the ones that have been previously inserted in the set. If the current solution is non-dominated then it is added, otherwise, it is discarded, and a new initial solution is constructed. It is important to mention that, to accelerate the computation time, the case for which the current solution would dominate any of the previous is not evaluated. As a result, the set of CS must contain at most $MaxIter$ different solutions. To finalize the procedure of obtaining the non-dominated Pareto set (F_0), a mechanism of obtaining the final set of non-dominated solutions is implemented. As mentioned above, since Pareto Candidate Set Insertion only evaluates if any of the previous inserted solutions dominates the current solution but not vice-versa, this procedures compares all of the solutions in the set to determine which ones belong to the non-dominated front.

5 COMPUTATIONAL RESULTS

To evaluate the proposed approach, we modified a set of benchmark instances originally proposed by Augerat et al. (Augerat et al., 1995) (P-instances) and Christofides and Elion (Christofides and Eilon, 1969) (E-instances) and also used in (Bruni et al., 2020). In those instances, we incorporated the information denoting whether a customer is mandatory to be visited or not. We have considered the general expression

$$\Gamma = \sqrt{\frac{\alpha}{1-\alpha}}$$

for the risk measure, with different value of $\alpha = 0.1, 0.5, 0.9$, to model different risk aversion levels. The algorithm was coded in C++ and the experiments were executed using a PC Intel®Core™i7 @2.30 GHz with 16 GB of RAM Memory under Windows 10 as OS. To account for the the randomness of the algorithm, it was ran 10 times per instance, considering different seed values at each execution and $MaxIter = 50$. To evaluate the performance of the algorithm, three quality multiobjective metrics were used:

- Number of points on the Pareto-Front (NPF) (Schott, 1995; Van Veldhuizen, 1999): This metric determines the ability to provide more choices for the decision-maker. The larger, the better.
- The k-nearest neighbor density estimation technique (k-D) (Zitzler et al., 2001). This metric allows to estimate the density of the fronts. In this work, the three density estimator is used. The smaller, the better.
- The Hypervolume of the space covered (Zitzler and Thiele, 1999). The main idea behind this metric is to compute the area of objective function space covered by the nondominated vectors. This metric estimates the size of the global dominated set in objective space. The larger, the better.

Table 1: Values of four quality metrics over E-instances (10 executions per instance) with a value of $\alpha = 0.1$.

Instance	Average			
name	NPF	k-D	Hypervolume	CPU time
En22k4	17.00	0.096	0.275	0.413
En23k3	10.30	0.179	0.394	0.382
En30k3	1.80	0.000	0.000	1.426
En30k4	3.80	0.361	0.470	1.736
En33k4	8.90	0.214	0.379	2.531
En51k5	15.40	0.119	0.247	18.679
En76k7	17.30	0.111	0.297	95.412
En76k8	20.50	0.087	0.235	109.466
En76k10	17.30	0.099	0.230	137.650
En76k14	26.20	0.065	0.221	173.438

Table 2: Values of four quality metrics over E-instances (10 executions per instance) with a value of $\alpha = 0.5$.

Instance	Average			
name	NPF	k-D	Hypervolume	CPU time
En22k4	18.60	0.080	0.226	0.417
En23k3	9.80	0.203	0.400	0.376
En30k3	1.50	0.000	0.000	1.415
En30k4	1.00	0.000	0.000	1.722
En33k4	8.20	0.233	0.358	2.530
En51k5	16.70	0.101	0.244	18.524
En76k7	18.10	0.098	0.293	95.498
En76k8	20.70	0.089	0.203	109.600
En76k10	18.40	0.102	0.319	137.543
En76k14	29.20	0.057	0.204	173.426

Additionally, we evaluate the elapsed CPU time in seconds to determine the effectiveness of the algorithm in finding solutions within a reasonable computational time.

Tables 1 to 6 show the average values for the proposed metrics. Column 1 indicates the name of the instance, while columns 2, 3 and 4 report the average valued for the above-mentioned metrics. The last column reports the elapsed time measured in seconds. Specifically, Tables 1 to 3 and 4 to 6 report the results for different values of α (0.1, 0.5 and 0.9, correspondingly). The reason of considering these values is to determine the ability of the algorithm to model the risk aversion.

According to the information shown in Tables 1–3, the algorithm produced dense fronts for all the values of α and good values of the hypervolume. To graphically display the performance of the algorithm over the group of E-instances, the ones with the minimum and maximum sizes were selected (En22k4 and En76k14, respectively) considering the front with the maximum number of points for each value of α for comparison. As can be observed, in both cases, the value of $\alpha = 0.9$ provides fronts with highest values of reward (and minimum values of risk) whereas, for the

Table 3: Values of four quality metrics over E-instances (10 executions per instance) with a value of $\alpha = 0.9$.

Instance	Average			
name	NPF	k-D	Hypervolume	CPU time
En22k4	15.00	0.093	0.252	0.413
En23k3	9.30	0.209	0.361	0.386
En30k3	2.70	0.577	0.634	1.441
En30k4	9.70	0.234	0.529	1.706
En33k4	8.20	0.266	0.482	2.554
En51k5	16.50	0.108	0.256	17.648
En76k7	19.60	0.093	0.298	95.793
En76k8	22.00	0.082	0.218	109.917
En76k10	17.90	0.101	0.256	137.707
En76k14	38.50	0.046	0.150	193.409

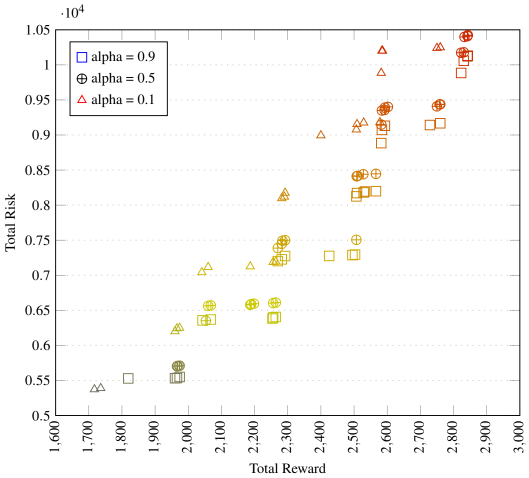

Figure 1: Pareto front for instance En22k4.

case when $\alpha = 0.1$, the values of risk are the largest. Thus, it can be concluded that, the parameter α has an effect over the quality of the solutions.

Tables 4, 5 and 6 summarizes the results over the P-instances.

For the set of P-instances, the algorithm showed a similar behavior as for the E-instances. Regarding the number of points, it is clear that the parameter α has not effect. Similarly, as the size of the instance increases, the average values of k-distances reduce, indicating that the algorithm produces dense fronts. With respect to the hypervolume, there is not a clear conclusion about the quality of the results.

Figures 3 and 4 show the results obtained for instances Pn16k2 and Pn76k5. As it can be noticed, for the instance Pn16k2, all of the values of α provided very similar Pareto fronts, which indicates that solutions are not deteriorating as the value of α increases. On the contrary, for the instance Pn76k5, the value of $\alpha = 0.1$ produced the Pareto front with smallest values of reward and high values of risk. In the case of the hypervolume, the value of $\alpha = 0.1$ produced the smallest values in average.

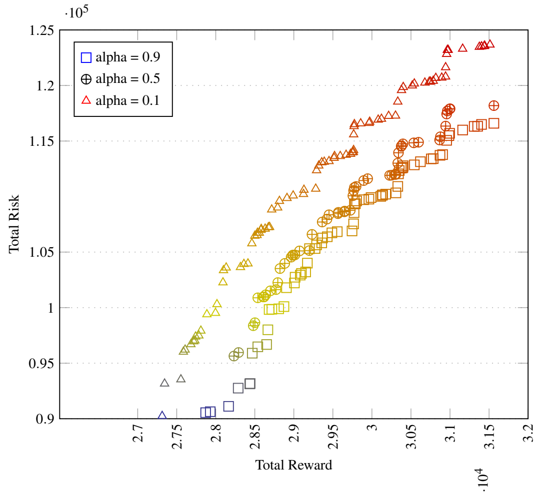

Figure 2: Pareto front for instance En76k14.

Table 4: Values of four quality metrics over P-instances (10 executions per instance) with a value of $\alpha = 0.1$.

Instance	Average			
name	NPF	k-D	Hypervolume	CPU time
Pn16k8	6.3	0.401	0.744	0.251
Pn19k2	4.7	0.518	0.857	0.210
Pn20k2	3.7	0.584	0.729	0.319
Pn21k2	6.4	0.330	0.578	0.348
Pn22k2	9.9	0.188	0.654	0.440
Pn23k8	12.4	0.154	0.764	1.794
Pn40k5	9.6	0.173	0.577	14.022
Pn45k5	12.4	0.150	0.750	24.161
Pn50k7	19.3	0.090	0.766	25.448
Pn50k8	9.9	0.160	0.646	26.636
Pn50k10	12.4	0.130	0.647	57.349
Pn51k10	12.5	0.125	0.772	35.037
Pn55k7	13.8	0.125	0.734	33.569
Pn55k8	12.3	0.156	0.796	37.299
Pn55k10	16.8	0.101	0.624	54.669
Pn60k10	19.6	0.087	0.754	65.975
Pn60k15	17	0.042	0.268	94.188
Pn65k10	23.3	0.046	0.265	93.541
Pn70k10	15.5	0.099	0.559	244.324
Pn76k4	9.5	0.173	0.614	80.482
Pn76k5	32.2	0.052	0.837	96.904

Regarding the CPU time, the algorithm showed a consistent performance over the E- and P-instances (increasing the execution time as the size of the instance increases).

6 CONCLUSIONS

In this work, we study a novel minimum latency bi-objective problem with profit collection and optional customers. For this problem a heuristic approach was proposed and developed for a broad class of risk measures. The performance was assessed using a set of

Table 5: Values of four quality metrics over P-instances (10 executions per instance) with a value of $\alpha = 0.5$.

Instance	Average			
name	**NPF**	**k-D**	**Hypervolume**	**CPU time**
Pn16k8	6.3	0.392	0.734	0.245
Pn19k2	4.4	0.653	0.994	0.209
Pn20k2	4	0.651	0.592	0.312
Pn21k2	6.1	0.309	0.671	0.346
Pn22k2	10.6	0.182	0.692	0.458
Pn23k8	12.1	0.167	0.859	1.770
Pn40k5	10.4	0.168	0.693	13.772
Pn45k5	13.1	0.137	0.631	24.801
Pn50k7	21.6	0.070	0.616	26.879
Pn50k8	11	0.168	0.784	29.604
Pn50k10	11.4	0.139	0.710	56.368
Pn51k10	11.3	0.157	0.770	35.650
Pn55k7	13.1	0.148	0.796	34.495
Pn55k8	12.7	0.154	0.793	37.371
Pn55k10	16.1	0.115	0.677	54.255
Pn60k10	20.8	0.076	0.672	66.105
Pn65k10	20.9	0.055	0.342	93.630
Pn70k10	16.5	0.102	0.637	188.625
Pn76k4	12.3	0.143	0.835	82.182
Pn76k5	35.6	0.051	0.896	95.832

Table 6: Values of four quality metrics over P-instances (10 executions per instance) with a value of $\alpha = 0.9$.

Instance	Average			
name	**NPF**	**k-D**	**Hypervolume**	**CPU time**
Pn16k8	7.5	0.324	0.775	0.237
Pn19k2	4.1	0.619	0.836	0.205
Pn20k2	3.3	0.715	0.688	0.325
Pn21k2	6.5	0.289	0.522	0.358
Pn22k2	12.6	0.162	0.811	0.433
Pn23k8	12.6	0.147	0.706	1.786
Pn40k5	10.2	0.184	0.655	13.765
Pn45k5	14.2	0.129	0.791	24.142
Pn50k7	21.1	0.083	0.790	26.536
Pn50k8	12.3	0.132	0.784	29.412
Pn50k10	12	0.141	0.733	56.085
Pn51k10	12.2	0.143	0.769	35.643
Pn55k7	13.7	0.124	0.798	33.561
Pn55k8	15.1	0.117	0.772	37.396
Pn55k10	19.4	0.088	0.716	56.543
Pn60k10	22.9	0.054	0.295	66.403
Pn60k15	1	–	–	93.891
Pn65k10	22.4	0.072	0.743	94.151
Pn70k10	18	0.094	0.633	188.721
Pn76k4	9.9	0.181	0.738	84.142
Pn76k5	34.4	0.050	0.852	95.949

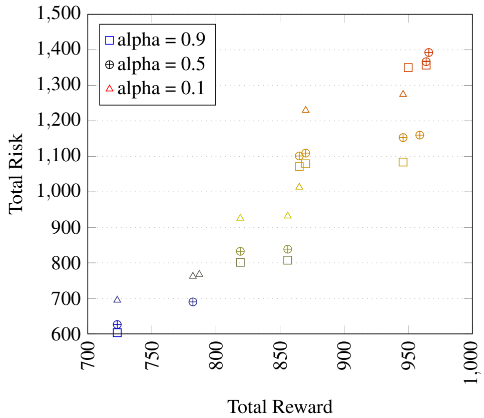

Figure 3: Pareto front for instance Pn16k8.

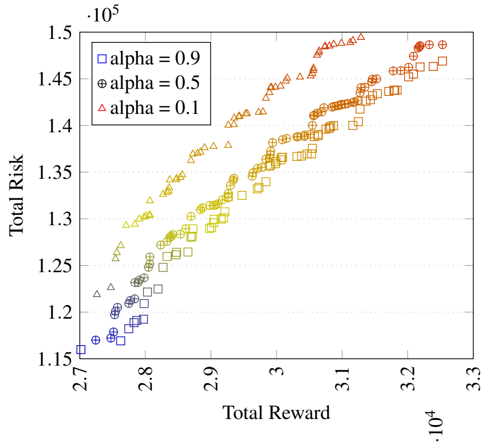

Figure 4: Pareto front for instance Pn76k5.

benchmark instances that were properly adjusted to analyze this particular problem. Specifically, the values of the multiobjective metrics indicate that the algorithm is able to find good quality fronts in a competitive computational time. The algorithm makes a positive contribution towards finding a trade-off between the profit and the risk aversion.

Future work can include the analysis of the case of capacitated vehicles or the inclusion of an objective function which is able to balance the maximum traveled distance among different vehicles, to provide equity in labor shifts. The incorporation of parameters such as time windows, or due dates for each node may also help to model real-life situations. In addition, including more objectives, mainly those related to environmental or social goals is an interesting research avenue.

ACKNOWLEDGEMENTS

This work was partially supported by the Universidad Panamericana through the grant ”Fondo Fomento a la Investigación UP 2019”, under project code UP-CI-2019-ING-GDL-08.

REFERENCES

Acerbi, C. (2002). Spectral measures of risk: A coherent representation of subjective risk aversion. *Journal of Banking & Finance*, 26(7):1505 – 1518.

Ahmadi-Javid, A. (2012a). Addendum to: Entropic value-at-risk: A new coherent risk measure. *J. Optimization Theory and Applications*, 155(3):1124–1128.

Ahmadi-Javid, A. (2012b). Entropic value-at-risk: A new coherent risk measure. *J. Optimization Theory and Applications*, 155(3):1105–1123.

Arellano-Arriaga, N., Alvarez-Socarras, A., and Martinez-Salazar, I. (2017). A sustainable bi-objective approach for the minimum latency problem. In *Alba E., Chicano F., Luque G. (eds) Smart Cities. Lecture Notes in Computer Science*.

Arellano-Arriaga, N., Molina, J., and Schaeffer, S. e. a. (2019). A two-phase metaheuristic for the cumulative capacitated vehicle routing problem. *J Heuristics*, 25(3):431–454.

Artzner, P., Delbaen, F., Eber, J.-M., and Heath, D. (1999). Coherent measures of risk. *Mathematical Finance*, 9(3):203–228.

Augerat, P., Belenguer, J., Benavent, E., Corberán, A., Naddef, D., and Rinaldi, G. (1995). Computational results with a branch and cut code for the capacitated vehicle routing problem research report 949-m. *Universite Joseph Fourier, Grenoble, France*.

Beraldi, P., Bruni, M. E., Laganà, D., and Musmanno, R. (2015a). The mixed capacitated general routing problem under uncertainty. *European Journal of Operational Research*, 240(2):382–392.

Beraldi, P., Bruni, M. E., Laganà, D., and Musmanno, R. (2019). The risk-averse traveling repairman problem with profits. *Soft Computing*, 23(9):2979–2993.

Beraldi, P., Bruni, M. E., Manerba, D., and Mansini, R. (2015b). A stochastic programming approach for the traveling purchaser problem. *IMA Journal of Management Mathematics*, 28(1):41–63.

Beraldi, P., Ghiani, G., Laporte, G., and Musmanno, R. (2005). Efficient neighborhood search for the probabilistic pickup and delivery travelling salesman problem. *Networks: An International Journal*, 45(4):195–198.

Bianco, L., Mingozzi, A., and Ricciardelli, S. (1993). The traveling salesman problem with cumulative costs. *Networks*, 23(2):81–91.

Bigras, L.-P., Gamache, M., and Savard, G. (2008). The time-dependent traveling salesman problem and single machine scheduling problems with sequence dependent setup times. *Discrete Optimization*, 5(4):685–699.

Bruni, M., Beraldi, P., and Khodaparasti, S. (2018a). A heuristic approach for the k-traveling repairman problem with profits under uncertainty. *Electronic Notes in Discrete Mathematics*, 69:221–228.

Bruni, M., Forte, M., Scarlato, A., and Beraldi, P. The traveling repairman problem app for mobile phones: a case on perishable product delivery. *AIRO Springer Series Volume "Advances in Optimization and Decision Science for Society, Services and Enterprises"*.

Bruni, M., Khodaparasti, S., and Beraldi, P. (2019). A selective scheduling problem with sequence-dependent setup times: A risk-averse approach. pages 1–7. SciTePress.

Bruni, M. E., Beraldi, P., and Khodaparasti, S. (2018b). A fast heuristic for routing in post-disaster humanitarian relief logistics. *Transportation research procedia*, 30:304–313.

Bruni, M. E., Beraldi, P., and Khodaparasti, S. (2020). A hybrid reactive grasp heuristic for the risk-averse k-traveling repairman problem with profits. *Computers & Operations Research*, 115:104–854.

Bruni, M. E., Guerriero, F., and Beraldi, P. (2014). Designing robust routes for demand-responsive transport systems. *Transportation research part E: logistics and transportation review*, 70:1–16.

Christofides, N. and Eilon, S. (1969). An algorithm for the vehicle-dispatching problem. *Journal of the Operational Research Society*, 20(3):309–318.

Dewilde, T., Cattrysse, D., Coene, S., Spieksma, F. C., and Vansteenwegen, P. (2013). Heuristics for the traveling repairman problem with profits. *Computers & Operations Research*, 40(7):1700–1707.

Ezzine, I. O., Semet, F., and Chabchoub, H. (2010). New formulations for the traveling repairman problem. In *Proceedings of the 8th International conference of modeling and simulation*. Citeseer.

Fischetti, M., Laporte, G., and Martello, S. (1993). The delivery man problem and cumulative matroids. *Operations Research*, 41(6):1055–1064.

Li, J. Y.-M. (2018). Technical note—closed-form solutions for worst-case law invariant risk measures with application to robust portfolio optimization. *Operations Research*, 66(6):1533–1541.

Lucena, A. (1990). Time-dependent traveling salesman problem–the deliveryman case. *Networks*, 20(6):753–763.

Méndez-Díaz, I., Zabala, P., and Lucena, A. (2008). A new formulation for the traveling deliveryman problem. *Discrete applied mathematics*, 156(17):3223–3237.

Mladenović, N., Urošević, D., and Hanafi, S. (2013a). Variable neighborhood search for the travelling deliveryman problem. *4OR*, 11(1):57–73.

Mladenović, N., Urošević, D., and Hanafi, S. (2013b). Variable neighborhood search for the travelling deliveryman problem. *4OR*, 11(1):57–73.

Ngueveu, S. U., Prins, C., and Calvo, R. W. (2010). An effective memetic algorithm for the cumulative capacitated vehicle routing problem. *Computers & Operations Research*, 37(11):1877–1885.

Nucamendi-Guillén, S., Angel-Bello, F., Martínez-Salazar, I., and Cordero-Franco, A. E. (2018). The cumulative capacitated vehicle routing problem: New formulations and iterated greedy algorithms. *Expert Systems with Applications*, 113:315–327.

Nucamendi-Guillén, S., Martínez-Salazar, I., Angel-Bello, F., and Moreno-Vega, J. M. (2016). A mixed integer formulation and an efficient metaheuristic procedure for the k-travelling repairmen problem. *Journal of the Operational Research Society*, 67(8):1121–1134.

Potvin, J.-Y. and Rousseau, J.-M. (1993). A parallel route building algorithm for the vehicle routing and scheduling problem with time windows. *European Journal of Operational Research*, 66(3):331–340.

Ribeiro, G. M. and Laporte, G. (2012). An adaptive large neighborhood search heuristic for the cumulative capacitated vehicle routing problem. *Computers & operations research*, 39(3):728–735.

Rivera, J. C., Afsar, H. M., and Prins, C. (2015). A multi-start iterated local search for the multitrip cumulative capacitated vehicle routing problem. *Computational Optimization and Applications*, 61(1):159–187.

Salehipour, A., Sörensen, K., Goos, P., and Bräysy, O. (2011). Efficient grasp+ vnd and grasp+ vns metaheuristics for the traveling repairman problem. *4or*, 9(2):189–209.

Schott, J. R. (1995). Fault tolerant design using single and multicriteria genetic algorithm optimization. Technical report, AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH.

Silva, M. M., Subramanian, A., Vidal, T., and Ochi, L. S. (2012). A simple and effective metaheuristic for the minimum latency problem. *European Journal of Operational Research*, 221(3):513–520.

Van Eijl, C. (1995). *A polyhedral approach to the delivery man problem*. Department of Math. and Computing Science, University of Technology.

Van Veldhuizen, D. A. (1999). Multiobjective evolutionary algorithms: classifications, analyses, and new innovations. Technical report, AIR FORCE INST OF TECH WRIGHT-PATTERSONAFB OH SCHOOL OF ENGINEERING.

Yongliang, L., Jin-Kao, H., and Qinghua, W. (June 2019). Hybrid evolutionary search for the traveling repairman problem with profits. *Information Sciences*.

Zitzler, E., Laumanns, M., and Thiele, L. (2001). Spea2: Improving the strength pareto evolutionary algorithm. *TIK-report*, 103.

Zitzler, E. and Thiele, L. (1999). Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. *IEEE transactions on Evolutionary Computation*, 3(4):257–271.

APPENDIX

In order to formulate the problem, we present the multi-layer network proposed in the deterministic context (Nucamendi-Guillén et al., 2018; Nucamendi-Guillén et al., 2016), and successfully extended by (Bruni et al., 2018a) for the risk-averse variant. Let L be the set of levels $L = \{1, \cdots, r, \cdots, N\}$, where $N = n-k+1$, and each level includes a copy of all the customers amended also with the depot in levels from 2 to n. Each tour in the network is represented by a path that ends in the first level and starts in a copy of the depot in some level. In fact, the level number represents the position of the customer in the tour: the customer in the first level is the last in the tour, the customer in the second level is the last but one, and so on. Two distinct tours cannot visit the same customer, neither in the same level nor in different levels. The model variables are defined as follows. Let x_i^r be a binary variable that takes value 1 iff customer i is visited at level r (i.e. there are $r-1$ customers to be visited after in the same tour); otherwise, it is set to 0. If $x_i^r = 1$, we say that customer i is active at level r. Let y_{ij}^r be another binary variable that is set to 1 iff edge (i, j) is used to link customer i active at level $r+1$ with customer j active at level r; otherwise, it takes value 0.

The mathematical formulation is expressed as follows.

$$Max: z_1 = \Big(\sum_{j\in V'}\sum_{r=1}^{N} \pi_j y_{0j}^r + \sum_{i\in V'}\sum_{\substack{j\in V'\\ j\neq i}}\sum_{r=1}^{N-1} \pi_j y_{ij}^r\Big) \quad (1)$$

$$Min: z_2 = \Big(\sum_{j\in V'}\sum_{r=1}^{N} r\mu_{0j} y_{0j}^r + \sum_{i\in V'}\sum_{\substack{j\in V'\\ j\neq i}}\sum_{r=1}^{N-1} r\mu_{ij} y_{ij}^r\Big) -$$

$$\Gamma\sqrt{\sum_{j\in V'}\sum_{r=1}^{N} r^2\sigma_{0j}^2 y_{0j}^r + \sum_{i\in V'}\sum_{\substack{j\in V'\\ j\neq i}}\sum_{r=1}^{N-1} r^2\sigma_{ij}^2 y_{ij}^r} \quad (2)$$

$$\sum_{r=1}^{N} x_i^r \leq 1 \quad i \in M \quad (3)$$

$$\sum_{r=1}^{N} x_i^r = 1 \quad i \in O \quad (4)$$

$$\sum_{i\in V'} x_i^1 = K \quad (5)$$

$$\sum_{r=1}^{N}\sum_{j\in V'} y_{0j}^r = K \quad (6)$$

$$y_{0i}^N = x_i^N \quad i \in V' \quad (7)$$

$$\sum_{\substack{j\in V'\\ j\neq i}} y_{ij}^r = x_i^{r+1} \quad i \in V', r = 1,2,\ldots,N-1 \quad (8)$$

$$y_{0j}^r + \sum_{\substack{i\in V'\\ i\neq j}} y_{ij}^r = x_j^r \quad j \in V', r = 1,2,\ldots,N-1 \quad (9)$$

$$x_i^r \in \{0,1\} \quad i \in V', r = 1,2,\ldots,N \quad (10)$$

$$y_{0j}^r \in \{0,1\} \quad j \in V', r = 1,2,\ldots,N \quad (11)$$

$$y_{ij}^r \in \{0,1\} \quad i, j \in V', i \neq j, r = 1,2,\ldots,N-1 \quad (12)$$

The objective function z_1 in (1) maximizes the total revenue.The second objective function minimizes the risk associated with the given routes. In particular, following the general expression discussed in Section 3, it is evaluated as the sum of the expected arrival time at the nodes plus the standard deviation of the total arrival time multiplied by a parameter Γ. Both the terms can be derived by applying the standard formula of the expected value and variance of the sum of independent random variables. Constraints (3) ensure that the optional customers i are served at most once. Constraints (4) guarantee that the mandatory customers are served. Constraints (5) and (6) ensure that only K starting and ending edges are created, whereas constraints (7) - (9) satisfy connectivity requirements. Finally, constraints (10) – (11) show the nature of variables.