Volume 30 (1), pp. 19–40
http://orion.journals.ac.za

ORiON
ISSN 0529-191-X

Application of the multi-objective cross-entropy method to the vehicle routing problem with soft time windows

C Hauman[*] JF Bekker[†]

Received: 31 May 2012; Revised: 22 March 2013; Accepted: 5 June 2013

Abstract

The vehicle routing problem with time windows is a widely studied problem with many real-world applications. The problem considered here entails the construction of routes that a number of identical vehicles travel to service different nodes within a certain time window. New benchmark problems with multi-objective features were recently suggested in the literature and the multi-objective optimisation cross-entropy method is applied to these problems to investigate the feasibility of the method and to determine and propose reference solutions for the benchmark problems. The application of the cross-entropy method to the multi-objective vehicle routing problem with soft time windows is investigated. The objectives that are evaluated include the minimisation of the total distance travelled, the number of vehicles and/or routes, the total waiting time and delay time of the vehicles and the makespan of a route.

Key words: Vehicle routing, soft time windows, cross-entropy method, multi-objective optimisation.

1 Introduction

The vehicle routing problem (VRP) remains one of the most studied problems in the field of operations research. It has many real-world applications, but exact methods require a considerable amount of computational time. The multi-objective optimisation (MOO) version of the vehicle routing problem with time windows (VRPTW) is considered, with a focus on the VRP with soft time windows variant (VRPSTW). In this problem type, a number of vehicles have to provide a service to customers at different locations while adhering to constraints with regard to the capacity of the vehicle and the time window

[*] Department of Industrial Engineering, University of Stellenbosch, Private Bag X1, Matieland, 7602, South Africa.

[†] Corresponding author: Department of Industrial Engineering, University of Stellenbosch, Private Bag X1, Matieland, 7602, South Africa, email: jb2@sun.ac.za

http://dx.doi.org/10.5784/30-1-127

in which service should start. Although the problem has been considered as a multi-objective problem by a number of authors [3, 9, 12, 13, 16, 20, 21], the focus has primarily been on minimising the number of vehicles and the total travel distances. This paper considers these and other pairs of conflicting objectives. The research aims were twofold: (1) to show that the multi-objective cross-entropy method (MOO CEM) can be applied to the VRPTW and (2) to provide reference solutions to a new set of benchmark problems recently developed by Castro-Gutierrez *et al.* (2011). Since the benchmark set is new, and reference solutions could not be found at the time of writing, the solution sets presented in this paper may serve as a first reference set for OR practitioners.

In general MOO almost always returns a set of two or more good solutions as opposed to single-objective optimisation, in which the optimum is a single solution. The decision maker still has to choose a specific solution from a set of MOO solutions. This set is often referred to as the Pareto optimal set.

The paper is structured into the following sections: it starts with a brief overview of the VRPTW and its formulation followed in the next two sections by an overview of the field of MOO in vehicle routing, and the nature of the CEM and its application to combinatorial optimisation. The formulation of the multi-objective model and the basic structure of the algorithm are then explained, followed by the presentation of the results. The final section concludes the paper.

2 The vehicle routing problem with time windows

In logistics and distribution, decision makers are often faced with the problem of developing optimal routes for vehicles that service different customers. The vehicle routing problem is considered to be a variation of the travelling salesperson problem where one salesperson has to visit a certain number of cities before returning to the home city. Also termed the truck dispatching problem, Dantzig & Ramser (1959) considered it a generalisation of the travelling salesperson problem. In the vehicle routing problem, a number of vehicles need to be routed to geographically dispersed nodes or customers. In addition, vehicles have limited capacity, which places a restriction on the number of nodes that one vehicle can visit. The vehicles also perform a service at the different nodes. The VRP has evolved into different subproblems, for example the VRP with stochastic demand and the VRPTW, which are more realistic representations of real-world problems. The VRPTW has many applications, such as the routing of buses and trains, bank deliveries and postal deliveries.

The problem under consideration in this study is the VRPTW. In this problem, the time in which a vehicle may arrive to begin service at a certain node is limited to a certain time window. The VRP with soft time windows further implies that vehicles can arrive after the time window has closed, although this is often associated with a penalty cost for late arrival.

In the VRPTW, a set of vehicles with limited capacity is to be routed from a central depot to a set of geographically dispersed customers with known demands and predefined time windows (Tan *et al.* 2006).

Toth & Vigo (2002) summarise the concept of the VRPTW, as applicable to this paper, as follows.

1. Each route visits the depot vertex.
2. Each customer vertex is visited by exactly one route (within the specified time window).
3. The total demand of the customers visited by a route does not exceed the capacity of the separate vehicles.

The addition of time windows increases the complexity and computational intensity of the problem. The VRP is classified as an **NP**-hard problem and consequently the VRPTW is a constrained **NP**-hard problem. Taillard *et al.* (1997) claim that in the relaxation of soft time windows feasible solutions are easier to find as there are fewer hard constraints, but further state that this is countered by the way hard time windows in turn allow for infeasible solutions to be filtered out fairly quickly.

Kallehauge *et al.* (2005) define the VRPTW in mathematical terms with a fleet of vehicles, $\mathcal{V}$, a set of customers $\mathcal{C}$ and a directed graph $\mathcal{G}$. $\mathcal{N}$ is the set of vertices, $0, 1, \ldots, n+1$ with 0 and $n+1$ representing the depot (respectively the starting and returning depot). Define x_{ijk} as

$$x_{ijk} = \begin{cases} 1, & \text{if vehicle } k \text{ drives directly from vertex } i \text{ to vertex } j, \\ 0, & \text{otherwise.} \end{cases}$$

Further definitions are the capacity of each vehicle (C_k), the demand of each customer i (D_i), the cost (or distance) c_{ij} and time (t_{ij}) associated with each arc (i, j) where $i \neq j$. The time window $[a_i, b_i]$ is associated with each customer. In the case of hard time windows the vehicle must arrive at the customer before b_i and in the case of soft time windows a delay time is logged. If a vehicle arrives before the time window starts, it incurs a waiting time until a_i when the service can start. The variable s_{ik} denotes the time at which vehicle k starts service at customer i. This is defined for every vehicle k and customer i, but becomes irrelevant if vehicle k does not service customer i. It is assumed that the time window of the depot always starts at zero and no service is required, *i.e.* $s_{0k} = 0$, and the time back at the depot (although no service is required) is defined as $s_{(n+1)k}$.

The mathematical model of Kallehauge *et al.* (2005) is adopted for the multi-objective VRP with soft time windows as shown in formulation (1) – (5).

The objective of the optimisation model is to

$$\text{minimise } k, \tag{1}$$

$$\text{minimise } \sum_{k\in\mathcal{V}} \sum_{i\in\mathcal{N}} \sum_{j\in\mathcal{N}} c_{ij} x_{ijk}, \tag{2}$$

$$\text{minimise } \left(\max_{k} \left(s_{(n+1)k} \right) \right), \tag{3}$$

$$\text{minimise } \sum_{k\in\mathcal{V}} t_w^k, \quad \text{and} \tag{4}$$

$$\text{minimise } \sum_{k\in\mathcal{V}} t_d^k \tag{5}$$

subject to the constraints

$$\sum_{k\in\mathcal{V}}\sum_{j\in\mathcal{N}} x_{ijk} = 1, \qquad i\in\mathcal{C}, \tag{6}$$

$$\sum_{i\in\mathcal{C}} D_i \sum_{j\in\mathcal{N}} x_{ijk} \leq C_k, \qquad k\in\mathcal{V}, \tag{7}$$

$$\sum_{j\in\mathcal{N}} x_{ojk} = 1, \qquad k\in\mathcal{V}, \tag{8}$$

$$\sum_{i\in\mathcal{N}} x_{ihk} - \sum_{j\in\mathcal{N}} x_{hjk} = 0, \qquad h\in\mathcal{C}, \quad k\in\mathcal{V}, \tag{9}$$

$$\sum_{i\in\mathcal{N}} x_{i,n+1,k} = 1, \qquad k\in\mathcal{V}, \tag{10}$$

$$x_{ijk}(s_{ik} + t_{ij} - s_{jk}) \leq 0, \qquad i,j\in\mathcal{N}, \quad k\in\mathcal{V}, \tag{11}$$

$$\sum_{i\in s_{ik}<a_i} (a_i - s_{ik}) = t_w^k, \qquad k\in\mathcal{V}, \tag{12}$$

$$\sum_{i\in s_{ik}>b_i} (s_{ik} - b_i) = t_d^k, \qquad k\in\mathcal{V}, \tag{13}$$

$$x_{ijk} \in \{0,1\}, \qquad i,j\in\mathcal{N}, \quad k\in\mathcal{V}. \tag{14}$$

Five conflicting objectives are defined in objectives (1)–(5) and optimised in pairs. The five objectives are shown in Table 1 with the labels defined by Castro-Gutierrez *et al.* (2011).

Objective	Label	Equation
Number of vehicles	Z1	(1)
Total travel distance	Z2	(2)
Makespan of tasks	Z3	(3)
Total vehicle waiting time	Z4	(4)
Total vehicle delay time	Z5	(5)

Table 1: *The objectives of the VRPSTW, posed by Castro-Gutierrez* et al. *(2011).*

The model differs from that of Kallehauge *et al.* (2005) in that the adaptation for soft time windows in (12) calculates the total time a vehicle waits for a time window to start on a route (t_w^k), and (13) calculates the total delay time of customers on a route waiting for vehicles that arrive after the close of a time window, denoted by t_d^k. The other constraints follow the original model. Constraint set (6) ensures each customer is visited exactly once, constraint set (7) is the capacity constraint and constraint set (14) ensures integrality. Constraint sets (8), (9) and (10) ensure that each vehicle leaves the depot, arrives at a customer and then proceeds to the next customer, and that all vehicles end at the depot. Sub-tour elimination constraints are excluded for brevity. This model with the soft time windows will be used as platform for further analysis.

3 Multi-objective optimisation and the VRP

The work of a few researchers in the field of MOO and the VRP is highlighted in this section. It is followed by a discussion of test problems in the research field.

The VRP is generally viewed as having the single objective of minimising the distance travelled by the different vehicles. However, over the past few years the problem has been considered as being multi-objective, with objectives including the distance travelled, average lateness and the number of vehicles. Jozefowiez *et al.* (2008) review the field of multi-objective vehicle routing problems. They identify three uses of multi-objective vehicle routing problems: an extension of classic academic problems in order to improve their practical applications, general classic problems and the study of real-life cases where the decision maker identified the objectives.

Jozefowiez *et al.* (2008) further group the methods used to solve the multi-objective problems into scalar methods, Pareto methods and a third category that considers different objectives separately. While it is possible to generate a weighted cost function with regard to two of the objectives in order to use scalar methods, this would generally result in a bias towards one of the objectives. The field of MOO has been developed to display the trade-off between objectives in an objective way. When considering the variation of the VRP with time windows, it appears that in the field of Pareto methods, evolutionary algorithms are used in most cases. Tan *et al.* (2006) propose a hybrid multi-objective evolutionary algorithm and Ombuki *et al.* (2006) use a multi-objective genetic algorithm. Geiger (2008) states that the relaxation of the time window restriction (VRPSTW) allows for a more practical multi-objective formulation and investigates the influence of this relaxation and other problem characteristics on genetic operators in evolutionary algorithms. Recently Garcia-Najera & Bullinaria (2011) proposed an improved multi-objective evolutionary algorithm which uses a similarity measurement to enhance the diversity of solutions. When compared to general evolutionary methods such as the popular NSGA-II (Deb 2001) this method shows improvements in particular in preserving high diversity before settling on a solution. Garcia-Najera & Bullinaria (2011) further demonstrated that 26 instances of Solomon's benchmark set exhibit conflicting objectives, and emphasized the use of a Pareto front.

In addition to providing algorithms for solving the VRPTW, Solomon (1987) also developed six sets of benchmark problems that have since been used in comparing different methods. Although the Solomon benchmark set has been extended and used in most literature on multi-objective vehicle routing, Castro-Gutierrez *et al.* (2011) found that classic test instances such as these problems developed by Solomon are not entirely suitable for MOO. The objectives used by Garcia-Najera & Bullinaria (2011) (the number of vehicles and the total travel distance) were in fact found to be in harmony for most of Solomon's problems. The need for specific multi-objective test cases as opposed to the extension of traditional single-objective cases came to light and a set of problem instances were generated by Castro-Gutierrez *et al.* (2011) to address this need. Initial experiments with the evolutionary algorithm NSGA-II (Deb 2001) showed evidence of multi-objective features such as a correlation value close to -1 or 1 for a given pair of objective values. In the case of minimisation, a correlation value close to -1 indicates a pair of objectives with a conflicting nature, *i.e.* the minimisation of one objective leads to an increase in the other objective. The newly proposed MOO test set was studied and thus provide some reference solution sets for researchers wishing to do further work in MOO of the VRPSTW. The MOO CEM was used, as it was recently proposed (Bekker & Aldrich 2010). Castro-Gutierrez *et al.* (2011) identified five objectives to be used in vehicle routing (see Table 1): the number of vehicles ($Z1$), the total travel distance ($Z2$), the makespan (travel time of longest route) ($Z3$), the total waiting time when vehicles arrive before the time window ($Z4$) and the total delay time when vehicles arrive after the time window ($Z5$).

4 Combinatorial optimisation and the cross-entropy method

Combinatorial optimisation refers to a problem where the decision maker seeks the combination of integer variable values that will optimise the objective function. Due to the combinatorial nature of the VRP, it is clear that the computational complexity of the problem increases as the number of vertices increases. The CEM is presented as being able to find a good local minimum for such a combinatorial optimisation problem. Rubinstein & Kroese (2004) devotes a chapter to the travelling salesperson problem (TSP). De Boer *et al.* (2005) also use the TSP as an example when illustrating the application of importance sampling to combinatorial optimisation. When comparing the CEM to other algorithms for combinatorial optimisation (simulated annealing, nested partitioning, tabu search, genetic algorithms), Rubinstein (1999) states that the CEM employs a global rather than a local search procedure. The interested reader is referred to Rubinstein & Kroese (2004) and Bekker & Aldrich (2010) for the mathematics supporting the CEM. A few basic concepts from the discrete case are repeated here for convenience (Rubinstein & Kroese 2004).

Let $\boldsymbol{x} \in \mathcal{X}$ be a random vector with probability mass function $f(\cdot, \boldsymbol{u})$ and distribution parameter vector $\boldsymbol{u}$. Suppose a problem has a performance function $S(\boldsymbol{x})$ with $x \in \mathcal{X}$ then the optimisation problem is

$$\text{maximise } S(\boldsymbol{x}) \text{ over } \boldsymbol{x} \in \mathcal{X}, \tag{15}$$

while its estimation formulation is to

$$\text{maximise } \hat{S}(\boldsymbol{x}) \text{ over } \boldsymbol{x} \in \mathcal{X}. \tag{16}$$

Suppose the maximum of S over $\mathcal{X}$ is γ^*, then

$$S(\boldsymbol{x^*}) = \gamma^* = \max_{\boldsymbol{x} \in \boldsymbol{\mathcal{X}}} S(\boldsymbol{x}). \tag{17}$$

The CEM requires that an estimation problem be associated with the optimisation problem of equation (17), and a collection of indicator functions $\{I_{\{S(\boldsymbol{x}) \geq \gamma\}}\}$ are defined on $\mathcal{X}$ for different values of $\gamma \in \mathbb{R}$. Let $\{f(\cdot, \boldsymbol{v}), \boldsymbol{v} \in \mathcal{V}\}$ be a family of probability mass functions (pmfs) on $\mathcal{X}$ that are parameterised by a real-value vector $\boldsymbol{v}$.

To solve the problem associated with equation (17), assume $\boldsymbol{u} \in \mathcal{V}$ and estimate the probability

$$l = \mathbb{P}_{\boldsymbol{u}}\{S(\boldsymbol{X}) \geq \gamma\} = \sum_{\boldsymbol{x} \in \boldsymbol{\mathcal{X}}} I_{\{S(\boldsymbol{x}) \geq \gamma\}} f(\boldsymbol{x}; \boldsymbol{u}) = E_{\boldsymbol{u}} I_{\{S(\boldsymbol{X}) \geq \gamma\}} \tag{18}$$

with $f(\boldsymbol{x};\boldsymbol{u})$ being the pmf on $\mathcal{X}$ and γ some chosen level. Suppose now γ is equal to γ^*, then $l = f(\boldsymbol{x}^*;\boldsymbol{u})$, which is a very small probability, and with this is associated a *rare event*. The probability can be estimated with the variance reduction technique of *Importance Sampling* (Rubinstein & Kroese 2004) by taking a random sample $\boldsymbol{X}_1,\ldots,\boldsymbol{X}_{\boldsymbol{N}}$ from a different pmf g and estimate l via

$$\hat{l} = \frac{1}{N}\sum_{k=1}^{N} I_{\{S(\boldsymbol{X}_k)\geq\gamma\}}\frac{f(\boldsymbol{X}_k;\boldsymbol{u})}{g(\boldsymbol{X}_k)}, \tag{19}$$

which is the unbiased *importance sampling estimator* of l. The distribution $g(\cdot)$ is introduced to make the probability of event l occurring less rare. The optimal manner to estimate l is to use the change of measure with a different pmf, namely

$$g^*(\boldsymbol{x}) := \frac{I_{\{S(\boldsymbol{x})\geq\gamma\}}f(\boldsymbol{x};\boldsymbol{u})}{l}. \tag{20}$$

Since this optimal pmf is generally difficult to obtain and depends on the unknown l, one approximates it by choosing g such that the *cross-entropy* or *Kullback-Leibler distance* between g and g^* is minimal. The Kullback-Leibler distance between two pmfs g and h is defined as

$$\begin{aligned} \mathcal{D}(g,h) &= \mathbb{E}_g\left[\log\frac{g(\boldsymbol{X})}{h(\boldsymbol{X})}\right] && (21)\\ &= \sum_{\boldsymbol{x}} g(\boldsymbol{x})\log g(\boldsymbol{x}) - \sum_{\boldsymbol{x}} g(\boldsymbol{x})\log h(\boldsymbol{x}). && (22)\end{aligned}$$

Since $I_{\{S(\boldsymbol{x})\geq\gamma\}}$ is non-negative, and the pmf f (of $\boldsymbol{X}$) is parameterised by a finite dimensional vector $\boldsymbol{v}$, *i.e.* $f(\boldsymbol{x}) = f(\boldsymbol{x};\boldsymbol{v})$, $g(\boldsymbol{x}) = f(\boldsymbol{x};\boldsymbol{v})$ for some reference parameter $\boldsymbol{v}$ (Kroese 2010). To estimate l with equation (19), $\boldsymbol{v}$ is chosen such that $\mathcal{D}(g^*, f(\cdot;\tilde{\boldsymbol{v}}))$ is minimal.

For discrete random vectors $\boldsymbol{X}$ the components of $\tilde{\boldsymbol{v}}$ will always be of the form (Alon *et al.* 2005)

$$\frac{\mathbb{E}_{\boldsymbol{v}}I_{\{S(\boldsymbol{X})\geq\gamma\}}I_{\{\boldsymbol{X}\in A\}}}{\mathbb{E}_{\boldsymbol{v}}I_{\{S(\boldsymbol{X})\geq\gamma\}}I_{\{\boldsymbol{X}\in B\}}} \tag{23}$$

with $A \subset B \subset \mathcal{X}$. This number can be estimated by taking a random sample $\boldsymbol{X}_1,\ldots,\boldsymbol{X}_N$ from the pmf $f(\cdot,\boldsymbol{v})$ and evaluate

$$\frac{\sum_{k=1}^{N} I_{\{S(\boldsymbol{X}_k)\geq\gamma\}}I_{\{\boldsymbol{X}_k\in A\}}}{\sum_{k=1}^{N} I_{\{S(\boldsymbol{X}_k)\geq\gamma\}}I_{\{\boldsymbol{X}_k\in B\}}}. \tag{24}$$

In the case of combinatorial optimisation problems such as the TSP and VRP, the probability distribution used in the importance sampling step is supported by a transition probability matrix. A matrix P is generated so that the probability of going from city i to city j is represented as p_{ij}. The method starts with equal probabilities in the matrix, which are updated according to the routes in the current best solution set. This is defined by De Boer *et al.* (2005), in the case of the travelling salesperson problem, as

$$p_{ij} = \frac{\sum_{l=1}^{N} I_{\{S(\boldsymbol{X}_l)\leq\gamma\}} I_{\{\boldsymbol{X}_l \in \mathcal{X}_{ij}\}}}{\sum_{l=1}^{N} I_{\{S(\boldsymbol{X}_l)\leq\gamma\}}}. \tag{25}$$

The estimated value for γ is obtained by creating a random sample of possible tours with evaluated objective function values. These values are then ranked to estimate a sample upper quantile, for example $\hat{\gamma} = S_{\varrho N}(\mathbf{X}_i)$ and typically, $0.70 \leq \varrho \leq 0.95$. The target event to be estimated by the method is denoted by $S(\boldsymbol{X}_l) \leq \gamma$ in the case of minimisation (De Boer *et al.* 2005). In equation (25), the probability of visiting city j (the jth success) is updated by counting the solutions in the current set of good solutions ($\boldsymbol{X}_l \leq \gamma$) with $\boldsymbol{X}_l \in \mathcal{X}_{ij}$, where $\mathcal{X}_{ij}$ is the set of matrices where the transition from i to j is made ($x_{ij} = 1$), in all the tours (l) generated as a population of size N in an iteration. In equation (25), only routes with a performance value less than γ are considered. Rubinstein & Kroese (2004) recommend that the updated matrix P be smoothed to help prevent premature convergence, using $P_t \leftarrow \alpha P_t + (1-\alpha)P_{t-1}, \quad 0 < \alpha < 1$.

The CE algorithm for optimisation of the VRP is based on Rubinstein & Kroese (2004), and is as follows.

1. *Initialise the transition matrix* $\widehat{P}_0$. *Typically, off-diagonal elements are assigned the value* $1/(n+1)$ *for* n *vertices. Initialise the iteration counter* $t \leftarrow$ *1.*
2. *Generate a sample of tours by some method using* P_{t-1}.
3. *Compute the estimation* $\hat{\gamma}_t = S_{\varrho N}(\boldsymbol{X}_i)$ *from the sample performances.*
4. *Update* $\widehat{P}_t$ *using equation* (25) *and the results of the previous step.*
5. *Apply smoothing:* $\widehat{P}_t \leftarrow \alpha\widehat{P}_t + (1-\alpha)\widehat{P}_{t-1}$.
6. *If* $\widehat{\gamma}_t$ *has converged, stop, otherwise* $t \leftarrow t = 1$ *and go to Step 2.*

Bekker & Aldrich (2010) adapted the CEM for MOO and tests on benchmark problems showed satisfactory results as approximations of the true Pareto sets were obtained with a relatively low number of simulations. The method was applied to deterministic, continuous problems and discrete, stochastic problems. In this study, the applicability of the method on discrete combinatorial problems is considered by maintaining the core mechanism of the CEM. The multi-objective model formulation is explained in §5.

The expression in equation (25) is applicable to the single-objective optimisation as applied by Rubinstein & Kroese (2004) and De Boer *et al.* (2005) with the set of best solutions (the elite set) defined as instances with a objective function value ($S(\boldsymbol{X}_l)$) smaller than γ. The ranking of the elite set in the case of MOO as proposed by Bekker & Aldrich (2010) and the subsequent construction of the probability matrix as used in this study are explained in §5 with the formulation of the optimisation model.

Ma (2011) succesfully applied the CEM for single-objective optimisation to the VRPTW. A multi-agent environment was introduced where a vehicle-specific transition matrix is associated with each node of the network that is used to construct a feasible route for every vehicle (the agents in this case). It is this transition matrix that is constructed using a random mechanism. It is then updated according to the performance of the routes travelled by the vehicles, in effect increasing the probability of estimating the rare event of an optimum solution. The proposed method employs the same principle, but uses a general transition matrix to construct routes for all the vehicles.

Ma (2011) further proposed a local search procedure on a subset of good solutions to avoid premature convergence by the CEM. The results on classic MOO test-instances by Bekker & Aldrich (2010) suggest that this is not necessary, but further research can investigate the influence of a local search procedure on the proposed multi-objective method in vehicle routing.

Chepuri & Homem-de-Mello (2005) applied the CEM with Monte Carlo sampling to the VRP with stochastic demand, using one vehicle. Generally, in this type of problem, there is variation in the set of customers visited, the demands and the travel times. Their application does not completely depend on specific problem formulations and can be extended to using multiple vehicles.

The applications of the CEM to the VRP are all based on single-objective optimisation. A MOO application using the CEM is presented next.

5 Model formulation

The VRPSTW consists of a network of customers at different locations indicated by coordinates, distances and travel times between customers and a central depot from which a homogenous fleet of vehicles depart. The solution consists of a list of the routes travelled by the different vehicles, usually illustrated by the order in which vertices are visited. The problem was modelled with the following data structures: customer detail, distance and travel time between pairs of customers, and the proposed routes (candidate solutions).

The pseudo-code for the MOO CEM for solving the VRPSTW is shown in Algorithm 1, and the structure of the optimisation model is shown in Figure 1. An initial probability matrix (P) with equal probabilities is defined to start the process. The probabilities are then used to construct routes in order to obtain a set of N solutions, with q the identifier (index) of the particular set of routes. This construction of routes and normalisation of the probability matrix is summarised in Algorithm 2. The solutions obtained in this manner are then evaluated and values are assigned to the pair of objectives being studied (in Figure 1, for Z_1 and Z_2). The solutions were ranked using the algorithm of Goldberg (1989), and the ranking value is stored in ρ_q. The degree of domination of one vector relative to the other vectors in the solution set is indicated by this value. All vectors with $\rho_q \leq t_h$ after ranking of an iteration are weakly dominated and form part of the elite set of that iteration. The value of t_h is a preset threshold value, typically $0 \leq t_h \leq 2$. If $t_h > 0$, then weakly and non-dominated vectors are returned in the elite set. If $t_h = 0$, then only non-dominated vectors are added to the elite set. This elite set of solutions is used to obtain the updated probability matrix as defined by (25) and explained in §4. The expression in equation (25) is adapted to

$$p_{ij} = \frac{\sum_{q=1}^{N} I_{\{\rho_q=0\}} I_{\{x_{ij}=1\}}}{\sum_{q=1}^{N} I_{\{\rho_q=0\}}}, \tag{26}$$

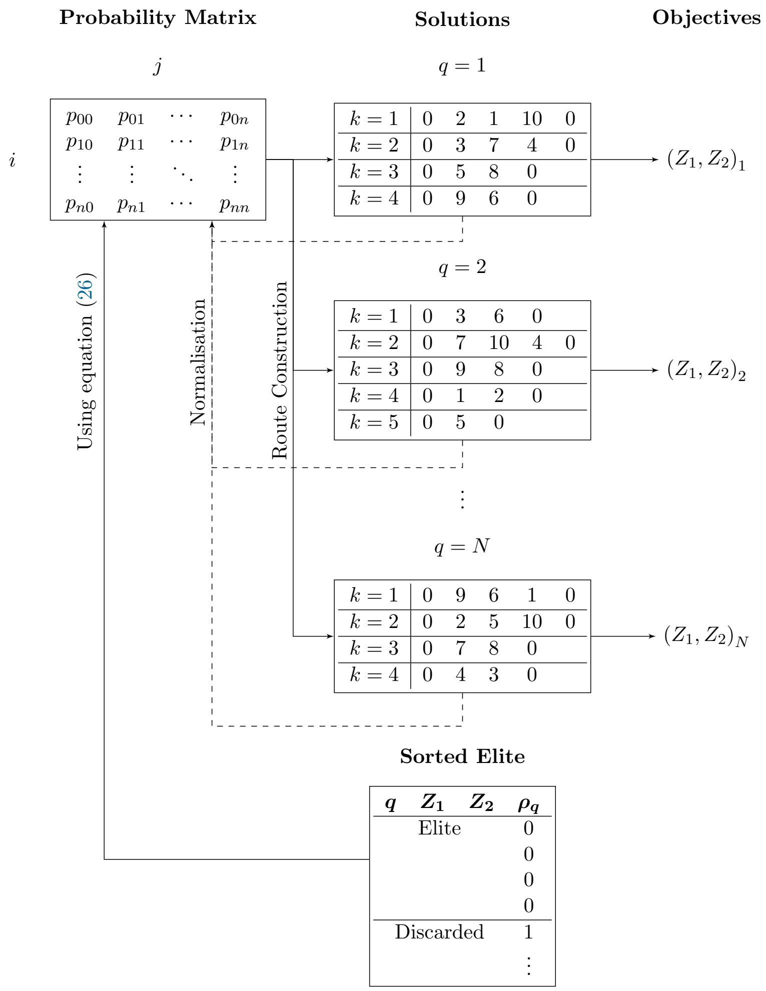

Figure 1: *A schematic representation of the structure of the optimisation model.*

to allow for the multi-objective case shown. Indicator values for every solution (q) with a number of routes (k) are summed in the population (N) as opposed to equation (25) where a solution consisted of one tour. This process is iterated until the stopping criterion is met.

Expression (26) can be explained with reference to Figure 1. Suppose the solutions with indices 1, 2 and N are the three solutions of the elite. The denominator will be equal to 3, while the various x_{ij} must be considered in the numerator. It can be seen that x_{09} occurs in all three solutions, so $p_{09} = {}^3\!/\!_3 = 1$, but p_{02} occurs in two of the three solutions, so $p_{02} = {}^2\!/\!_3$.

The algorithm applied to the vehicle routing problem with time windows is illustrated in Algorithm 1, and is based on Bekker & Aldrich (2010).

For the purpose of Algorithm 1 it is important to note that a rank of 0 denotes non-domination and accordingly the best solutions of the current set. The loop of lines 6 to 13 retains solutions with a rank of 2 to maintain solution diversity and to prohibit premature

Algorithm 1: MOO VRPSTW with the CEM

```
1  Let $P$ be the transition matrix of probabilities, $N_m$ the maximum number of loops, $\tau$ the maximum
   number of evaluations per loop and $\boldsymbol{\mu}$ the set of averages of the objectives in the elite set;
2  $L_c \leftarrow 0$;
3  repeat
4      $t \leftarrow 0$;
5      Initialise $P_t$ and $\boldsymbol{\mu}_t$;
6      repeat
7          $t \leftarrow t + 1$;
8          Construct routes using Algorithm 2 and $P_t$;
9          Evaluate routes;
10         Rank the solutions using the threshold $t_h = 2$;
11         Update $P_t$ using equation (26);
12         Smooth $P_t \leftarrow \alpha P_t + (1 - \alpha)P_{t-1}$;
13     until $|\boldsymbol{\mu}_t - \boldsymbol{\mu}_{t-1}| \leq \epsilon$ or $t \geq \tau$;
14     Rank with $t_h = 1$;
15     $L_c \leftarrow L_c + 1$;
16 until $L_c > N_m$;
17 Rank with $t_h = 0$;
18 return [elite set];
```

convergence. The non-dominated solutions are only isolated at the end of the algorithm, in line 17.

In principle, routes are constructed with Markov-chain transition probabilities but in reality the VRP is highly constrained and routes are dependent on the feasibility of said transition. Algorithm 2 ensures that the transition to non-feasible vertices is not possible by setting the transition probability to 0 and normalising the remaining probabilities. The matrix with transition probabilities is updated according to feasible vertices to be visited. From the array of feasible cities the next city to be visited is visited according to the row of P.

6 Methods and results

The algorithm for the CEM was coded in MATLAB 7.10 and applied to the different benchmark instances of Castro-Gutierrez *et al.* (2011).

6.1 Benchmark problems

Castro-Gutierrez *et al.* (2011) generated a benchmark problem set that may be used for the multi-objective VRPSTW. The time windows and the demands of the customers are characterised in a certain way. The problems are characterised with the number of customers, the different time window profiles (0–4, 4 being the tightest) and capacity and/or demand constraints (0–2, 2 being the tightest). Each problem has a specific label, for example, "50_d0_tw1" denotes the benchmark problem with 50 customers, a demand/capacity profile number 0 and time window profile 1. Six pairs of objectives (from the five defined in Table 1) were found to be conflicting in the datasets for soft time windows, namely $(Z1, Z3)$, $(Z1, Z5)$, $(Z2, Z3)$, $(Z2, Z5)$, $(Z4, Z3)$ and $(Z4, Z5)$ (Castro-Gutierrez *et*

Algorithm 2: Construction of routes of the VRPSTW

```
1  Let depot be customer $i + 1$ (next customer);
2  while all customers are not visited do
3      Previous customer $i$ = Customer $i + 1$;
4      for all customers not yet visited do
5          feas$_1$ ← (capacity of truck);
6          feas$_2$ ← (time window (in the case of hard time windows));
7          feas$_3$ ← (vehicle is back at the depot in time);
8          Feasible ← feas$_1$ and feas$_2$ and feas$_3$;
9          if Not Feasible then
10             Change probability to be visited from current customer to 0;
11     if No customers are feasible then
12         The next customer is the depot;
13         Increment routes $k \leftarrow k + 1$;
14         return to Step 2;
15     else
16         Calculate the new weighted row of the probability matrix;
17         Sample a customer from this row;
18         Add this customer to the route as customer $(i + 1)$;
```

al. 2011). These objectives are used in the application of the MOO CEM. Instances with negative correlation values (close to -1) for all pairs of objectives were used, *i.e.* for all demand/capacity profiles and for the time window profiles (1–4) as the time window profile 0 only exhibits a time window for the depot and the problem reduces to a general VRP (no waiting time or delay time).

6.2 Performance measures

The hypervolume comparison method yields a recognised unary indicator used in comparing two different Pareto sets in order to assess the difference in quality of two algorithms. A unary indicator in this context is a function that returns a single, real value from the set $\mathbb{R}$. The hypervolume indicator (I_H) is also the only unary indicator capable of detecting that a set of solutions is not worse than another (Zitzler *et al.* 2003, Raad *et al.* 2011). This indicator is used to isolate the best result of 10 pseudo-independent tests of the MOO CEM for the VRP. To show the general performance of the method, the highest, lowest and average values for I_H are documented for the 10 runs. Since pairs of objectives are considered in this study, the hyperarea indicator instead of the hypervolume indicator is used.

The indicator values of all problem sets on which the algorithm was tested are shown in Table 2. It is important to note that due to the difference in the order of the objectives, the hyperarea is a relational indicator, *i.e.* a high value in one column is incomparable to the seemingly lower value in another column. It is also interesting to note that in some cases (especially in the $Z1$ and $Z5$ pairing) the hyperarea indicator is equal to 0 for all the documented indicator values of particular problem cases. This shows that the algorithm performed the same for all 10 runs as there was no difference in the hyperarea of the final approximate Pareto fronts. This is applicable to discrete cases such as the number of vehicles ($Z1$) and in the case of $Z4$ and $Z5$ when an optimal solution (such as delay time = 0 or waiting time = 0) is found in all 10 iterations. In these cases the movement of the second objective is irrelevant as the hyperarea indicator depends on an area, and here the extreme point of this objective forms a line with the secondary objective, thus resulting in an indicator value of 0. Variances higher than 0.5 are indicative of answers that did not exhibit a final multi-objective front (*i.e.* one optimal point) but the best iteration provided a solution better than the other iterations, resulting in a large variation in the hyperarea indicator value.

6.3 Parameter setting

One objective of the research is to determine if the MOO CEM can, in general, be used for the VRPSTW, so the fine-tuning of parameters is not the main focus. However, before tests were performed on the benchmark problems, experiments were conducted to get an indication of good parameter values. The main parameters that influence results include the population size N, the smoothing parameter value of α, and lastly the maximum allowed number of iterations (τ and N_m), as explained in Algorithm 1.

A number of experimental tests were conducted and the averages of the I_H indicator of five runs were computed from a common reference. In Figure 2 the population size was set at $N = 2\,000$ and the average I_H of five runs at different values of α showed $\alpha = 0.9$ performed the best. In Figure 2, the value of $\alpha = 0.8$ was set while different values of N were tested. The difference in the indicator values for $N = 2\,000$ and $N = 2\,500$ was deemed not large enough to warrant the increase in computational time associated with $N = 2\,500$. Following these experiments and due to time considerations, tests were conducted with parameters set to $N = 2\,000$, $\alpha = 0.9$, $\tau = 25$ and $N_m = 10$.

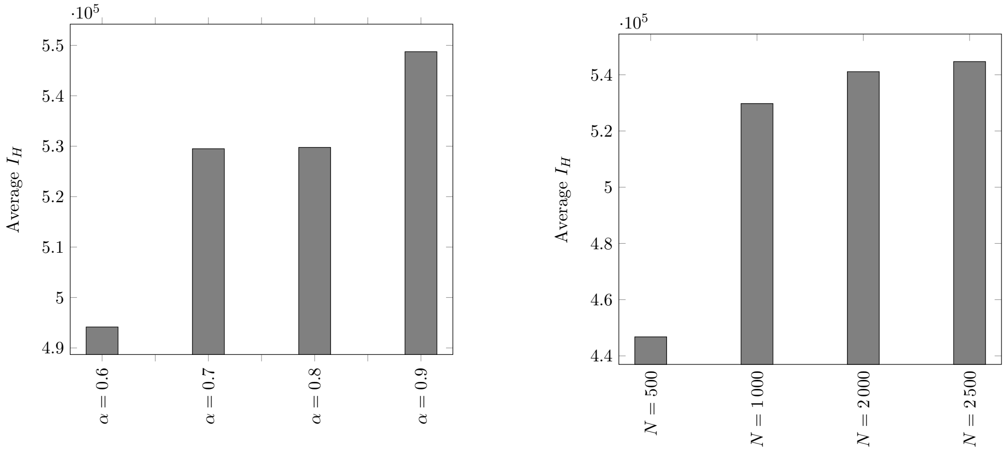

Figure 2: *Variation of I_H for different values of α and N, respectively.*

6.4 Results

Tests were performed on all cases of the 50 customer problems with conflicting objectives, but this paper addresses only one problem instance in detail in this section, namely

Test instance		Z1-Z3	Z1-Z5	Z2-Z3	Z2-Z5	Z4-Z3	Z4-Z5
50_d0_tw1	best	2160	25380	413334	3857736	69721200	0
	average	1704	4716	375580.8	2645854	62858160	0
	worst	1380	0	343026	1641396	53330400	0
50_d0_tw2	best	3120	0	287310	1030632	88077600	291600
	average	2964	0	258177	621264.6	80239680	150120
	worst	2820	0	230874	63354	73411200	0
50_d0_tw3	best	300	0	477456	21774401	11361600	2.21E+08
	average	258	0	402969.6	19475492	9065880	1.58E+08
	worst	240	0	320574	18388050	6595200	87138000
50_d0_tw4	best	3780	158880	229692	61954032	89737200	89737200
	average	3258	137802	194004.6	53912637	81240840	79200720
	worst	2760	99420	158562	48499344	63478800	35424000
50_d1_tw1	best	3000	17160	443322	1810542	58820400	0
	average	2730	5148	386695.8	1297729	50774040	0
	worst	2520	0	317346	948810	43167600	0
50_d1_tw2	best	1740	0	235548	2880042	86929200	0
	average	1560	0	188147.4	1557728	78718680	0
	worst	1440	0	148326	268500	72554400	0
50_d1_tw3	best	1080	0	366858	25042092	18925200	2.55E+08
	average	768	0	324304.2	24065212	16047360	2E+08
	worst	540	0	268722	23141700	12146400	1.22E+08
50_d1_tw4	best	2460	20340	687600	5750988	75999600	1.66E+08
	average	2094	11178	576431.4	3960852	70805880	1.55E+08
	worst	1680	0	496482	2033028	61135200	94795200
50_d2_tw1	best	0	123060	831528	13276242	59760000	61689600
	average	0	122784	694749.6	10267123	49678920	33022080
	worst	0	120300	594426	8175126	36532800	0
50_d2_tw2	best	2640	0	713970	3056874	66567600	284400
	average	2484	0	630510	2398305	59207760	145440
	worst	2220	0	551298	1336314	44132400	0
50_d2_tw3	best	420	87360	490368	57652992	7732800	25596000
	average	342	87096	458721	49416319	5318280	15331680
	worst	240	86040	376998	51309015	2350800	0
50_d2_tw4	best	2400	0	963108	2949780	82825200	14742000
	average	2226	0	900396	1458745	50275080	5660280
	worst	2100	0	821442	600840	29865600	0
250_d2_tw1	best	600	1033920	107118	4.72E+08	11491200	2.7143E+10
	average	252	845604	79979.4	4.09E+08	8209800	1.8608E+10
	worst	60	599640	53874	3.55E+08	4946400	1.3184E+10
250_d2_tw2	best	540	1029300	227088	4.9E+08	24958800	3.4807E+10
	average	276	820344	173223	4.44E+08	17012880	2.7858E+10
	worst	120	688980	112242	3.96E+08	10735200	2.3874E+10
250_d2_tw3	best	180	1173840	31140	5.1E+08	3661200	2.2704E+10
	average	96	944016	17226.6	4.87E+08	1946880	1.7887E+10
	worst	0	775740	0	4.54E+08	0	1.3895E+10
250_d2_tw4	best	420	1485540	89472	4.08E+08	25833600	2.5134E+10
	average	173.3	1281972	61505.33	3.82E+08	17179560	2.2787E+10
	worst	60	1072440	41910	3.34E+08	9165600	2.0649E+10

Table 2: *Hyperarea values of test problem results.*

50_d1_tw4. This problem instance is chosen as a fair representation of results found across the different demand and time windows, and serves as a good platform to discuss the performance of the algorithm and the findings of the study.

The results are shown in subsequent figures, as follows: for each of the conflicting pairs of objectives, a figure showing the progression of the approximation front through the iterations (or loops) as explained in Algorithm 1 is presented (see *e.g.* Figure 3(a)). This serves as indicator of the worth of the CEM, as results improve from the completely random construction of routes in iteration 1 to the final approximation front formed in iteration 10. Figure 3(b) shows the final approximation front in the objective domain as presented to the decision maker. Lastly, a table showing the routes for the set of vehicles ($\mathcal{V}_k$) of a particular solution point from the final approximation front, is presented. The tables (Tables 3 – 5) list the reference numbers of the customers in the order that they were visited by each vehicle. These numbers follow from Castro-Gutierrez *et al.* (2011). In Table 3, for example, it is shown that six vehicles were used, and vehicle number 1 ($V1$) visited eight customers. Also, there are exactly 50 non-zero labels in the table, and the label sets represented by each column are mutually exclusive. Solutions not listed may be obtained from the corresponding author.

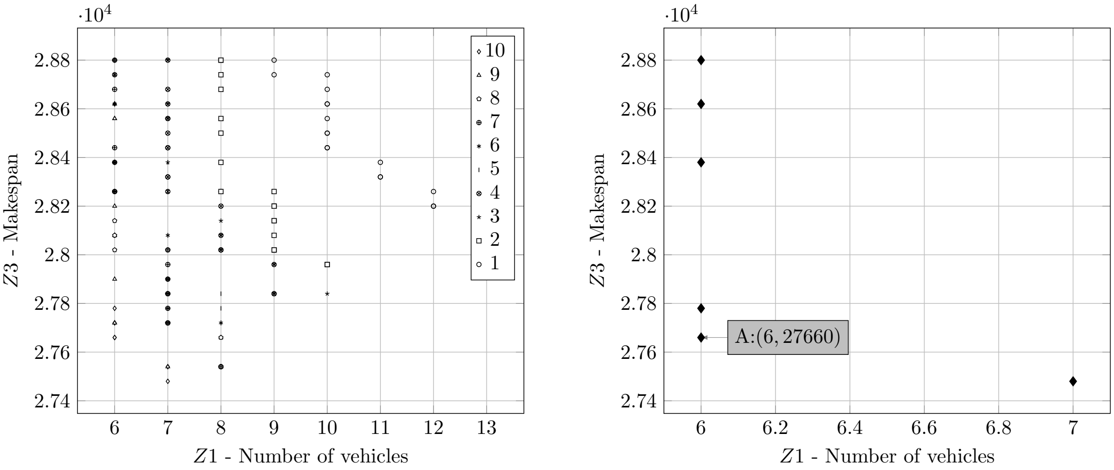

Figure 3: *Front progression and the final Pareto-front of test instance 50_d1_tw4 for $Z1$ vs $Z3$, respectively.*

6.5 Discussion of results

For all six cases the progression of the approximation front is evident and the worth of the CEM of estimating good solutions is illustrated. In the case of the discrete objective $Z1$, the number of vehicles, Figures 3 and 4 show approximate Pareto fronts that are limited in their multi-objective nature and exhibit seemingly redundant points at the lowest number of vehicles. As this is only evident in the case of the discrete objective, it can be deduced that the algorithm found a number of solutions for $Z3$ for the same value of $Z1$, and cannot justify fewer or more values for $Z1$, as no points between discrete values can be obtained. In Figure 3 for example, one can see that using six vehicles in this problem yield a number of makespan values, of which 27 660 time units is the least (Point A). The

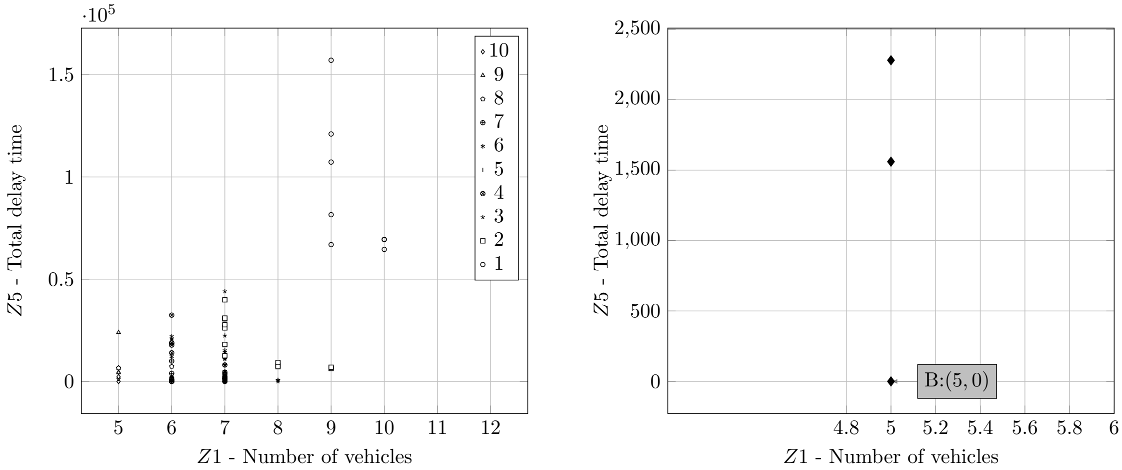

Figure 4: *Front progression and the final Pareto front of test instance 50_d1_tw4 for $Z1$ vs $Z5$, respectively.*

V1	V2	V3	V4	V5	V6
0	0	0	0	0	0
649	1 725	2 000	1 173	2 104	1 870
1 384	856	430 148	430 030	1 714	2 107
430 761	1 888	1 897	1 703	430 378	1 777
2 044	1 362	1 813	1 781	948	1 203
2 121	1 678	1 463	669	430 625	1 389
2 149	2 073	486	2 003	0	1 875
430 804	1 588	1 509	0		974
482	1 686	661			1 235
0	2 152	907			2 138
	106	430 471			0
	0	1 721			
		2 007			
		0			

V1	V2	V3	V4	V5
0	0	0	0	0
2 104	1 813	1 173	649	1 870
856	1 725	1 384	2 107	1 777
1 888	1 897	1 714	907	1 203
430 148	1 678	430 378	2 073	1 703
1 463	430 030	2 044	486	1 389
1588	974	1 875	1 509	669
106	430 761	1 235	1 686	482
661	1 781	2 121	2 152	0
430 471	2 149	2 138	1 362	
2 000	430 804	0	1 721	
430 625	2 003		0	
2 007	0			
948				
0				

Table 3: *Routes of solutions A and B, respectively, for the problem 50_d1_tw4 ($Z1$ vs $Z5$).*

makespan can be reduced to 27 480 time units if the decision maker is willing to buy seven vehicles. In Figure 4, the best solution is shown by Point B, where the delay time is zero when using five vehicles.

Although Figure 8 exhibits a single-point solution (which is strictly not Pareto anymore), Figure 8 illustrates that the non-optimal solutions are inherently multi-objective. The progression of the front to an optimal (the best combination for $Z4$ and $Z5$ is 0) is good, and the multi-objective nature of the objective pair is not refuted.

It is also valuable to note that in Figure 6 and to a lesser extent Figures 5, 7 and 8, the spread of the front becomes smaller as the algorithm progresses, and tends to converge to a limited region. This supports the finding that the number of loops (10) is adequate. If time permits, more outer loops may be added, but this would only lead to small improvements in the front. This choice is thus left to the decision maker, *i.e.* to what extent priority is given to the execution time of the algorithm or to the precision of the solution.

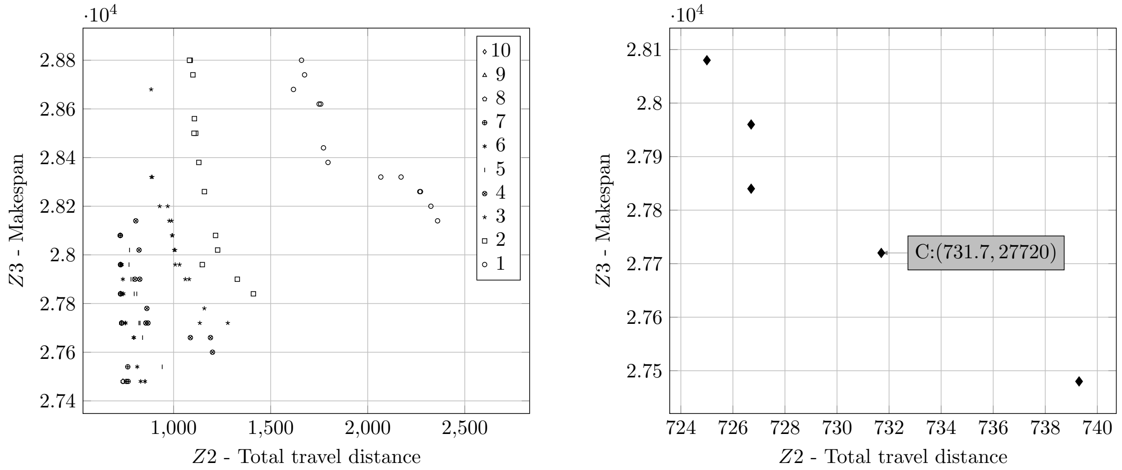

Figure 5: *The front progression and the final approximation front of test instance 50_d1_tw4 for $Z2$ vs $Z3$, respectively.*

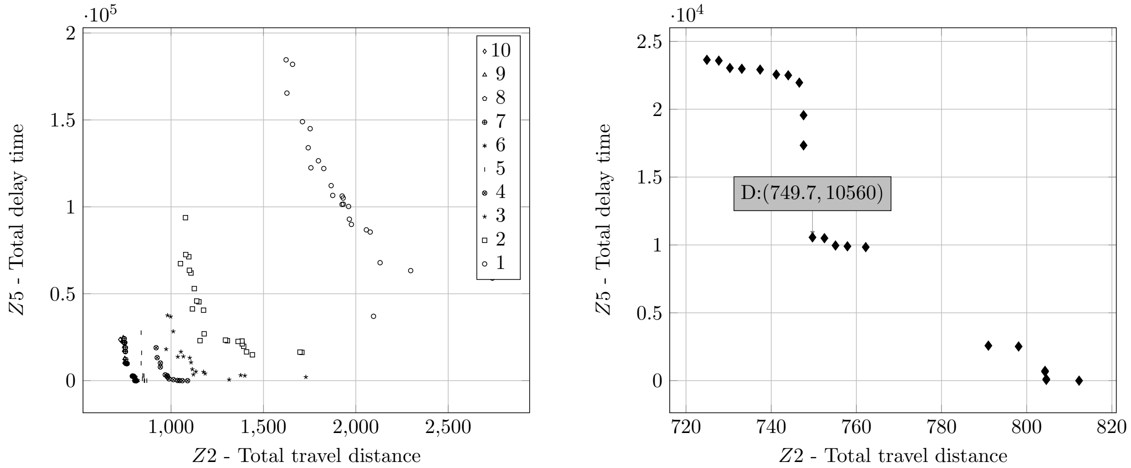

Figure 6: *The front progression and the final approximation front of test instance 50_d1_tw4 for $Z2$ vs $Z5$, respectively.*

The set of problems with the tightest capacity profile (250_d2_tw1), in the 250-customer problem, was computed to investigate the feasibility of the CEM for multi-objective vehicle routing on larger scale problems. The method proved to be able to solve these problems and results showed similiar characteristics to that of the 50-customer problems. The biggest difference can be seen in Figure 9: in the 250_d2 problem objective $Z3$ exhibited the same behaviour as that of the discrete objective $Z1$, the number of vehicles. This is due to the nature of the problem and time windows, limiting the makespan of the longest route to a fixed number of possibilities. Figure 10 serves as an example of the $Z4$ and $Z5$ pairing, exhibiting a multi-objective approximation front as also found in the other non-documented 50-customer problems.

MOO is generally descriptive and not prescriptive, and the decision maker still has to choose a solution for implementation from the approximation set. In the results set pre-

V1	V2	V3	V4	V5	V6
0	0	0	0	0	0
1 813	1 714	1 588	430 378	430 471	1 725
2 104	649	430 761	856	1 721	1 870
1 173	430 030	2 107	907	2 152	1 389
1 384	1 703	974	486	430 625	2 044
1 678	1 777	1 781	1 509	0	2 121
661	1 203	430 804	948		1 875
106	2 138	482	2 007		1 235
2 073	0	0	1 463		2 149
1 897			0		669
2 000					2 003
1 362					0
1 686					
430 148					
1 888					
0					

V1	V2	V3	V4	V5	V6
0	0	0	0	0	0
1 725	1 870	1 714	2 104	856	1 588
2 107	2 044	430 378	1 813	1 173	1 463
430 761	2 121	106	1 384	649	2 073
1 781	1 389	661	1 897	430 030	430 625
669	1 235	486	1 888	1 703	2 000
482	1 875	907	430 148	1 777	0
0	974	1 509	1 686	1 203	
	2 149	1 678	2 152	2 138	
	430 804	2 007	1 721	0	
	2 003	948	1 362		
	0	430 471	0		
		0			

Table 4: *Routes of solutions C and D, respectively, for test instance 50_d1_tw4 (Z2 vs Z5).*

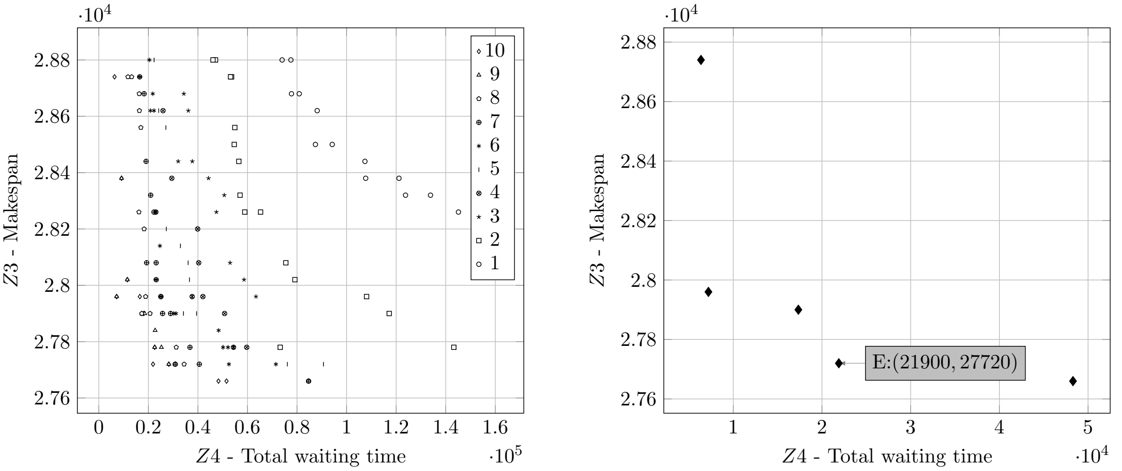

Figure 7: *The front progression and the final approximation front of test instance 50_d1_tw4 for Z4 vs Z3, respectively.*

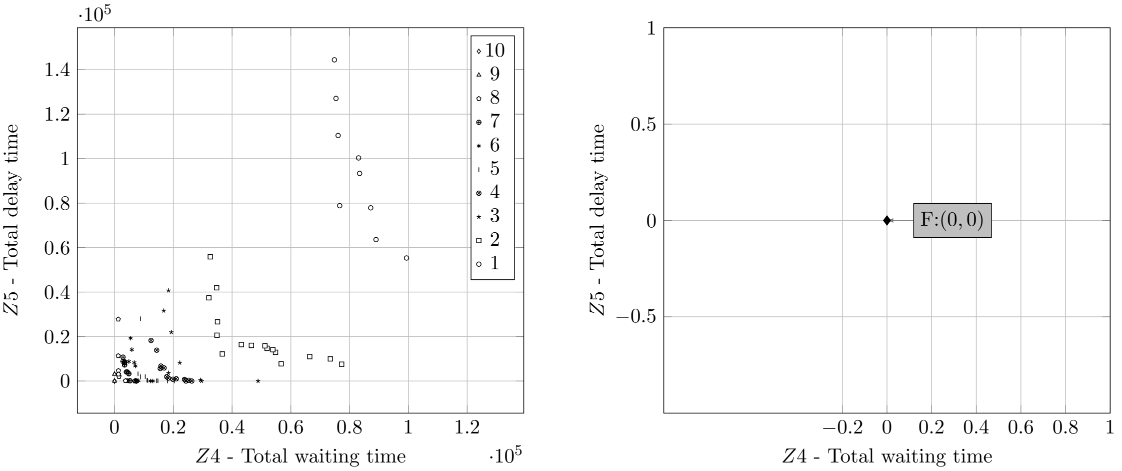

Figure 8: *The front progression and the final approximation front of test instance 50_d1_tw4 for Z4 vs Z5, respectively.*

V1	V2	V3	V4	V5	V6
0	0	0	0	0	0
2 107	2 104	1 725	1 777	649	1 870
1 897	430 378	1 888	2 000	1 714	430 030
1 703	2 121	974	2 044	1 678	1 389
1 235	907	1 588	1 686	1 362	1 509
2 149	856	1 781	2 152	1 384	1 721
430 804	669	2 138	106	486	430 625
482	2 003	0	0	2 073	430 471
0	0			948	0
				661	
				2 007	
				1 463	
				0	

V1	V2	V3	V4	V5	V6
0	0	0	0	0	0
649	2 107	1 725	1 870	1 897	1 173
430378	1 777	1 384	2 000	1 813	856
1203	106	2 104	1 875	1 714	1 888
661	1 463	1 703	2 044	430 030	430 761
2149	1 509	1 588	2 121	974	1 235
430471	907	669	1 686	1 389	2 073
430625	486	2 003	2 152	1 721	430 148
0	1 781	0	1 362	2 007	2 138
	430 804		0	948	0
	482			1 678	
	0			0	

Table 5: *Routes of solutions E and F, respectively, for test instance 50_d1_tw4 (Z4 vs Z3).*

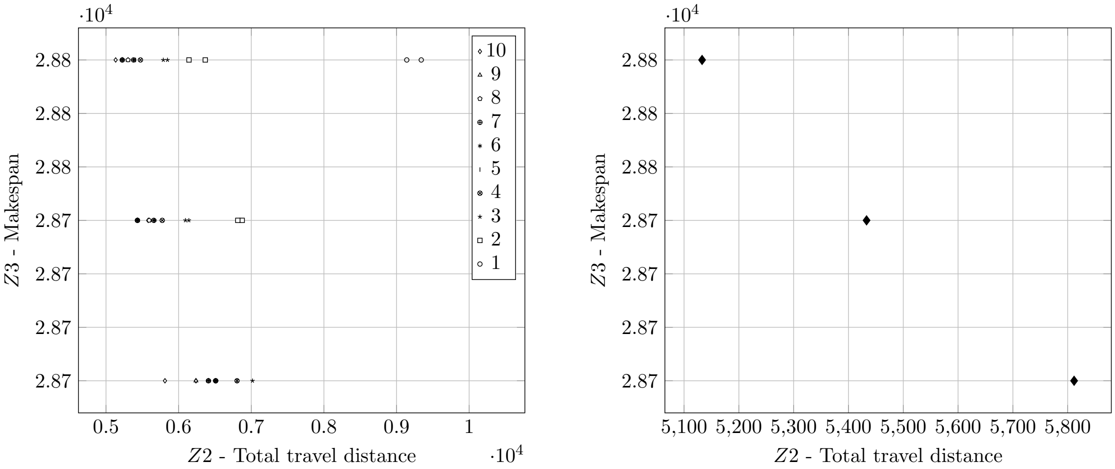

Figure 9: *The front progression and the final approximation front of test instance 250_d2_tw1 for Z2 vs Z3, respectively.*

sented it is in some cases easy to choose a solution (Figures 4 and 8), but in other cases additional business constraints will be required to support a choice. In Figure 4, six or seven vehicles can be bought; if finances are limited, then six vehicles are indicated, but the makespan is penalised. In Figure 6, suppose the business wants to limit travel distance due to high fuel cost, then solutions in the upper left region should be considered. Now the total delay time increases, while a good trade-off solution is perhaps indicated by Point D.

The discussion of the results is concluded with Figure 11 illustrating the nature of the routes on a coordinate map with the depot at $(28.0718, -16.622)$. The complex nature of the multi-objective VRPSTW is illustrated since a comparison of different objectives yields completely different routing solutions. Figures 12 and 13 are an illustration of the routes of one solution of the 250_d2_tw1 problem for objectives $Z4$ vs $Z5$. The solution consisted of 37 routes which are displayed over four figures of nine or ten routes each to provide clarity.

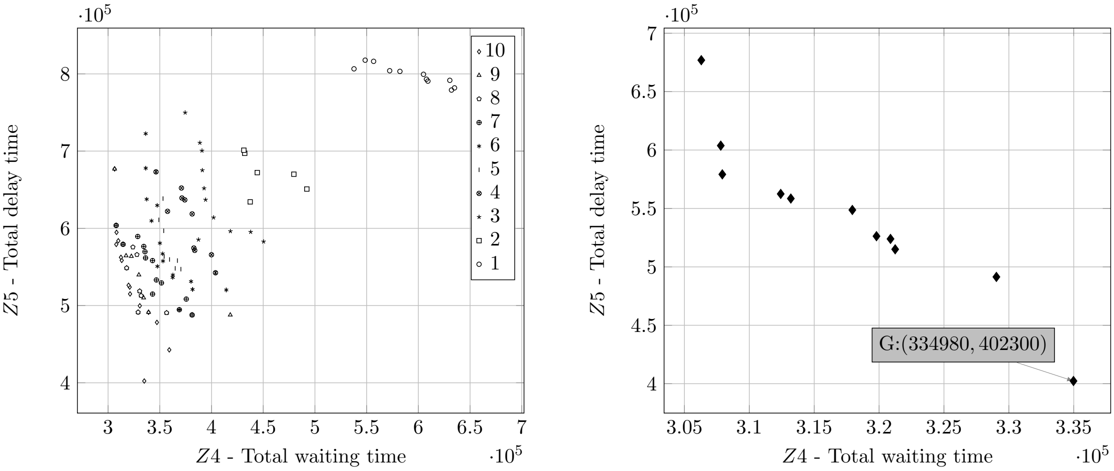

Figure 10: *The front progression and the final approximation front of test instance 250_d2_tw1 for Z4 vs Z5, respectively.*

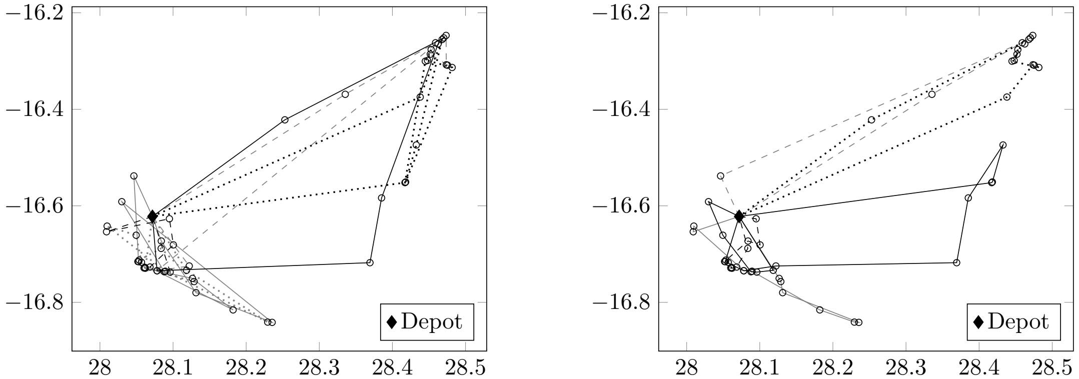

Figure 11: *Map of routes of solution A and F, respectively, for test instance 50_d1_tw4 for Z1 vs Z3.*

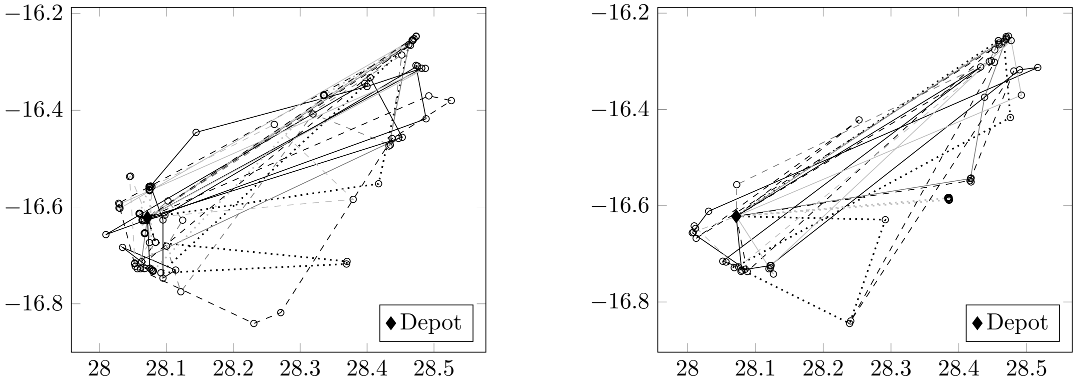

Figure 12: *Map of routes of solution G, Part 1 and Part 2, respectively, for test instance 250_d2_tw1 for Z4 vs Z5.*

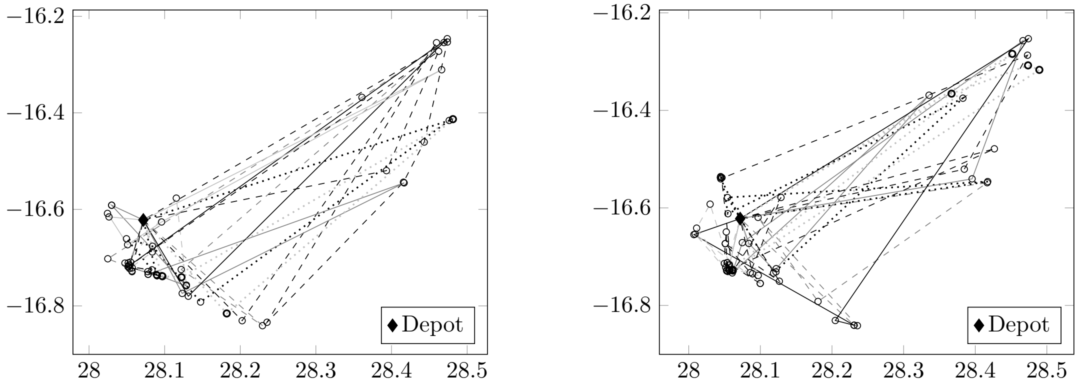

Figure 13: *Map of routes of solution G, Part 3 and 4, respectively, for test instance 250_d2_tw1 for Z4 vs Z5.*

7 Summary and conclusions

The first aim of this paper was to assess the possible application of the CEM for MOO to the multi-objective vehicle routing problem with soft time windows. It was found that the MOO CEM adequately obtains an approximation set that progresses towards the Pareto front of solutions for conflicting objectives of a given problem. Some of the benchmark problems converged to a final solution consisting of a single point in the objective space even though it initially consisted of a front of solutions.

The speed performance of the MOO CEM itself proved good but the route-construction algorithm (as explained in §2) is a definite bottleneck. Future work should look at streamlining the construction of routes with a transition probability matrix as required by the CEM.

Reference sets of solutions for MOO of VRPs with soft time windows are also provided that OR practitioners may use and improve upon.

Future work might assess the applicability of other metaheuristics to the Gutierrez *et al.* (2011) problem set, and to compare the performance of these algorithms.

References

[1] Alon G, Kroese DP, Raviv T & Rubinstein RY, 2005, *Application of the cross-entropy method to the buffer allocation problem in a simulation-based environment*, Annals of Operations Research, **134(1)**, pp. 137–151.

[2] Bekker J & Aldrich C, 2010, *The cross-entropy method in multi-objective optimisation: An assessment*, European Journal of Operationa; Research, **211(1)**, pp. 112–121.

[3] Castro-Gutierrez J, Landa-Silva D & Moreno Perez J, 2011, *Nature of real-world multi-objective vehicle routing with evolutionary algorithms*, IEEE International Conference on Systems, Man, and Cybernetics, Nottingham, pp. 257–264.

[4] Chepuri K & Homem-de-Mello T, 2005, *Solving the vehicle routing problem with stochastic demands using the cross-entropy method*, Annals of Operations Research, **134**, pp. 153–181.

[5] Coello Coello CA, 2006, *Evolutionary multi-objective optimization: A historical view of the field*, Computational Intelligence Magazine, IEEE, **1(1)**, pp. 28–36.

[6] Dantzig GB & Ramser JH, 1959,*The truck dispatching problem*, Management Science, **6(1)**, pp. 80–91.

[7] De Boer PT, Kroese D, Mannor S & Rubinstein R, 2005, *A tutorial on the cross-entropy method*, Annals of Operations Research, **134(1)**, pp. 19–67.

[8] Deb K, 2001, *Multi-objective optimization using evolutionary algorithms*, Wiley, Hoboken (NJ).

[9] Garcia-Najera A & Bullinaria JA, 2011, *An improved multi-objective evolutionary algorithm for the vehicle routing problem with time windows*, Computers & Operations Research, **38(1)**, pp. 287–300.

[10] Geiger MJ, 2003, *A computational study of genetic crossover operators for multi-objective vehicle routing problem with soft time windows*, pp. 191–207, in Habenicht W, Scheubrein B & Scheubrein R (Eds), *Multi-Criteria- und Fuzzy-Systeme in Theorie und Praxis*, Deutscher Universitats-Verlag, Frankfurt.

[11] Goldberg DE, 1989, *Genetic algorithms in search, optimization and machine learning*, Addison-Wesley Publishing Company, Boston (MA).

[12] Jozefowiez N, Semet F & Talbi EG, 2008, *Multi-objective vehicle routing problems*, European Journal of Operational Research, **189(2)**, pp. 293–309.

[13] Kallehauge B, Larsen J, Madsen OBG & Solomon MM, 2005, *Vehicle routing problem with time windows*, pp. 67–98 in Desaulniers G, Desrosiers J & Solomon MM (Eds), *Columns Generation*, Springer, New York (NY).

[14] Kroese DP, 2010, *Cross-Entropy Method* in Cchewochran JJ, Cox LA, Keskinocak P, Kharoufeh JP & Smith JC (Eds), *Wiley Encyclopaedia of Operations Research and Management Science*, John Wiley & Sons, New York (NY).

[15] Ma TY, 2011, *A cross entropy multiagent learning algorithm for solving vehicle routing problems with time windows*, pp. 59–73, in Böse JW, Hu H, Jahn C, Shi X, Stahlbock R, Voß S (Eds), *Computational Logistics*, Springer, New York (NY).

[16] Ombuki B, Ross BJ & Hanshar F, 2006, *Multi-objective genetic algorithms for vehicle routing problem with time windows*, Applied Intelligence, **24(1)**, pp. 17–30.

[17] Raad D, Sinske A & van Vuuren JH, 2011, *Water distribution systems design optimisation using metaheuristics and hyperheuristics*, ORiON, **27(1)**, pp. 17–44.

[18] Rubinstein R, 1999, *The cross-entropy method for combinatorial and continuous optimization*, Methodology and Computing in Applied Probability, **1(2)**, pp. 127–190.

[19] Rubinstein RY & Kroese DP, 2004, *The cross-entropy method: A unified approach to combinatorial optimization, monte-carlo simulation, and machine learning*, Springer, New York (NY).

[20] Solomon MM, 1987, *Algorithms for the vehicle routing and scheduling problems with time window constraints*, Operations Research, **35(2)**, pp. 254–265.

[21] Taillard ED, Badeau P, Gendreau M, Guertin F & Potvin JY, 1997, *A tabu search heuristic for the vehicle routing problem with soft time windows*, Transportation Science, **31(2)**, pp. 170–186.

[22] Tan KC, Chew YH & Lee LH, 2006, *A hybrid multiobjective evolutionary algorithm for solving the vehicle routing problem with time windows*, Computational Optimization and Applications, **34(1)**, pp. 115–151.

[23] Toth P & Vigo D, 2002, *The vehicle routing problem*, 9$^{\text{th}}$ Edition, Society for Industrial Mathematics, Philadelphia (PA).

[24] Zitzler E, Thiele L, Laumanns M, Fonseca CM & da Fonseca VG, 2003, *Performance assessment of multiobjective optimizers: An analysis and review*, Evolutionary Computation, IEEE Transactions on, **7(2)**, pp. 117–132.