Improved Dynamic Lexicographic Ordering for Multi-Objective Optimisation

Juan Castro-Gutierrez[1], Dario Landa-Silva[1], and José Moreno Pérez[2]

[1] ASAP Research Group, School of Computer Science, University of Nottingham, UK
jpc@cs.nott.ac.uk, dario.landasilva@nottingham.ac.uk
[2] Dpto. de Estadística, I.O. y Computación, Universidad de La Laguna, Spain
jamoreno@ull.es

Abstract. There is a variety of methods for ranking objectives in multi-objective optimization and some are difficult to define because they require information a priori (e.g. establishing weights in a weighted approach or setting the ordering in a lexicographic approach). In many-objective optimization problems, those methods may exhibit poor diversification and intensification performance. We propose the Dynamic Lexicographic Approach (DLA). In this ranking method, the priorities are not fixed, but they change throughout the search process. As a result, the search process is less liable to get stuck in local optima and therefore, DLA offers a wider exploration in the objective space. In this work, DLA is compared to Pareto dominance and lexicographic ordering as ranking methods within a Discrete Particle Swarm Optimization algorithm tackling the Vehicle Routing Problem with Time Windows.

Keywords: Multi-objective Optimization, Swarm Optimization, Combinatorial Optimization, Vehicle Routing Problem.

1 Introduction and Motivation

Multi-objective Optimization (MOO) problems have a number of objectives that are usually in conflict, so improving one objective leads to worsen another. In particular, *many-objective optimization problems* involve the optimization of four or more objectives, presenting a considerable challenge for some solutions methods.

Solution methods for multi-objective optimization differ mainly on the way they rank solutions. Many successful approaches exist to address this issue in problems with few objectives (i.e. less than four). However, these ranking schemes have not exhibited the same performance in *many-objective optimization problems*. Some classical ranking methods like Pareto dominance, use a strict ranking scheme that sometimes fails to discriminate between solutions, as it only accepts improvements in all objectives at the same time. On the other hand, methods like the lexicographic approach impose a more static behavior, as objectives are ranked according to a fixed relative importance.

In previous work [2] we introduced the Dynamic Lexicographic Approach (DLA). This is an alternative dynamic multi-objective ranking approach for

R. Schaefer et al. (Eds.): PPSN XI, Part II, LNCS 6239, pp. 31–40, 2010.

many-objective optimization. DLA offers an intuitive approach to establish a dynamic ranking among objectives. Rather than establishing a fixed priority among the objectives, the decision-maker establishes a preference. This preference is then used with a probability mass function (*pmf*) to generate a vector of priorities that changes dynamically throughout the search process.

We used DLA in [2] as a an additional mechanism to rank the quality of multi-objective solutions in a Particle Swarm Optimization approach applied to tackle the Solomon's instances of the Vehicle Routing Problem with Time Windows (VRPTW) treated as a MOO problem. We use the DLA to select the best leader in the neighborhood of each particle, while still using the traditional Pareto dominance to decide whether to update solutions. This combined approach (*Pareto + DLA*) was compared against the alternative of using Pareto dominance for both tasks (*Pareto + Pareto*): selection of the leader and update of solutions. Results indicated that (*Pareto + DLA*) was better than (*Pareto + Pareto*) in clustered problems *(cxxx)*. But the (*Pareto + DLA*) approach showed poor performance on random *(rxxx)* and random-clustered *(rcxxx)* instances.

This work investigates the role of the probability mass function (*pmf*) and the use of DLA to both select the leader and choose the solution to update. We propose two versions of DLA, one using a greedier function to assign probabilities to each preference and the other using two phases involving two probability functions. For k iterations, one probability mass function is used to encourage intensification. For the remaining iterations, another probability mass function is used to encourage diversification. The two DLA versions are compared to Lexicographic ordering and Pareto dominance on well-known instances of the VRPTW using the hypervolume as performance metric. Our results indicate that the DLA is a valuable ranking alternative method for MOO.

This paper is organized as follows. The basics of multi-objective optimization are given in Section 2. The algorithm for the Dynamic Lexicographic Approach is detailed and exemplified in Section 3 while Section 4 describes the Particle Swarm Optimization method focusing on those elements important to our research. A brief description of the Vehicle Routing Problem with Time Window is given in Section 5. We describe our experiments in Section 6 and discuss results in Section 7. Finally, our contribution and proposed further research are summarized in Section 8.

2 Multi-Objective Optimization (MOO)

In MOO, we aim to solve a problem of the type: *minimize* $\boldsymbol{f}(\boldsymbol{x}) = f_1(\boldsymbol{x})$, $f_2(\boldsymbol{x})$,...,$f_k(\boldsymbol{x})$, subject to: $g_i(\boldsymbol{x}) \leq 0, i = 1, 2, ..., m$ and $h_i(\boldsymbol{x}) = 0,\ i = 1, 2, ..., p$. Where the decision variable vector is $\boldsymbol{x} = [x_1, x_2, ..., x_n]^T$, each objective function f_i is defined in $\Re^n \rightarrow \Re, i = 1, 2, ..., k$ and the constraint functions are g_i and $h_i, i = 1, 2, ..., m, j = 1, 2, ..., p$ which are defined in the same domain as f_i.

Pareto Dominance is based on the concept of dominance. Without loss of generality, we consider a minimization problem. A vector $\boldsymbol{u}$ is said to *dominate* $\boldsymbol{v}$ (denoted by $\boldsymbol{u} \prec \boldsymbol{v}$), iif $\forall i \in (1, ..., k) : u_i \leq v_i \wedge \exists i \in (1, ..., k) : u_i < v_i$.

Moreover, we say that a vector in the feasible region ($\boldsymbol{u} \in \mathcal{F}$) is *Pareto Optimal*, if there is not any other vector in that region ($\boldsymbol{u'} \in \mathcal{F}$), such that $\boldsymbol{u'}$ *dominates* $\boldsymbol{u}$ ($\boldsymbol{u'} \prec \boldsymbol{u}$). We define the *Pareto Front* as the set of vectors in $\Re^n$, such that all elements in the set are *Pareto Optimal*.

Using Pareto dominance in many-objective optimization has an implicit pitfall derived from its manner to discriminate solutions. This method accepts solutions in a linear fashion that inhibits the worsening of objectives, which sometimes provokes the search to get stuck. Recent approaches have extended this ranking method using the concept of *relaxed dominance or α-dominance*. This type of dominance, proposed by Kokolo et al. [9], tries to overcome the weakness above explained by allowing the worsening in some objective if there are improvements in others. However, it might be difficult to set the appropriate bounds to get a desirable performance.

Lexicographic ordering is another widely used ranking method in which the decision-maker establishes fixed preferences among objectives. Given a priority order (a, b), the objective whose priority is a is compared in first place and the one with b in second. This way, we can formally describe the lexicographic comparison as: $(a, b) \leq (a', b')$ *iif* $a < a'$ or ($a = a'$ and $b \leq b'$). The same procedure can be easily extended to k priorities. This technique works well when the number of objectives is small. When dealing with a large number of priorities, objectives with low priority will not be likely to be compared and therefore, could be ignored in practice. Moreover, the fixed order, in which the method discriminates vectors, makes the search liable for premature convergence [3].

In addition to Pareto dominance and lexicographic ordering, there exist many other ranking methods to discriminate solutions in multi-objective scenarios. A review on these methods can be found in the work of Ehrgott and Gandibleux [7].

3 Dynamic Lexicographic Approach

The Dynamic Lexicographic Approach (DLA) offers an intuitive approach to establish a dynamic ranking among objectives. The decision-maker establishes preferences among objectives and a function to associate a probability to each preference. These probabilities are used to create different vectors defining a standard lexicographic ordering each time. This approach overcomes the limitation in the number of objectives because DLA does not rule out any preference. Even preferences with a low probability have a chance to appear in the first position of the vector of priorities. Additionally, the continuous change of priorities makes it possible to avoid premature convergence as the exploration is broader.

To clarify how DLA works, we provide an example for $N = 4$ objectives. Lets assume that objective i is assigned preference i, that is, $pref(o_i) = i$ for $i = 0, \ldots, N-1$, where N is the number of objectives. Suppose the decision-maker provides the function $p(i) = 0.8exp(-0.4i)$. Firstly, we evaluate this function for $i = 0, \ldots, N-1$. Since $N = 4$, we calculate the probabilities as $p(0) = 0.8, p(1) = 0.54, p(2) = 0.36$ and $p(3) = 0.24$. Secondly, these probabilities are normalized as $p'(i) = p(i)/\sum_{k=0}^{N-1} p(k)$, obtaining the values $0.41, 0.28, 0.19, 0.12$. In a third

step, we split the segment $[0, 1]$ into sub-segments according to the accumulate probability. Therefore, we obtain $[0, 0.41, 0.69, 0.88, 1]$, such that each preference has a sub-interval assigned. The first preference has interval $[0, 0.41)$, the second $[0.41, 0.69)$, the third $[0.68, 0.88)$ and the fourth $[0.88, 1]$. The algorithm goes through the above steps only once. Regarding the generation of priority vectors, roulette-wheel selection is applied using this segment. First, a random number r is generated with uniform distribution in $[0, 1]$. Lets suppose that $r = 0.70$. As r falls in the sub-interval $[0.68, 0.88)$, the third objective will be pushed back in the vector of priority $v \leftarrow 3$. After this operation, we remove the selected sub-interval and the segment is re-scaled, and another random number is generated. After $N - 1$ times repeating this process, we get a priority vector that can be used in a lexicographic approach to discriminate solutions.

The probability mass function (*pmf*) plays a key role in the performance of the DLA. Depending on the shape of this function, different probability values are assigned to each objective producing different lexicographic orderings. Therefore, assuming that the decision-maker establishes decreasing preferences (as in the example above), there are three main types of functions: linear, quadratic and exponential. Linear functions assign probabilities with a constant step among them. Quadratic functions assign a similar probability to those objectives with the highest preferences and zero to those with the lowest. Finally, an exponential function assigns high and distinct probabilities to objectives with high preference and assigns low but non-zero probability to those with low preference.

4 Particle Swarm Optimization (PSO)

PSO is a swarm-based stochastic algorithm proposed by Kennedy and Eberhart [8]. A swarm is formed by a group of particles that moves on the search space. Each particle knows its current position x_i, its best position b_i achieved so far and the best position reached by the swarm so far g. Moreover, each particle i shares its current position with its neighbors n_i. Each particle possesses an independent velocity which is updated taking into account the position of the particle, that of its neighbors and that of the best positioned particles.

The PSO algorithm was originally proposed to tackle continuous optimization problems. For discrete optimization, a number of alternatives have been proposed re-interpreting how particles move. Here, we have adopted the approach proposed by Consoli et al. [4]. This algorithm lacks the use of velocity conceiving the move of particles by using the equation: $x_{i,j} \leftarrow x_i \vee g_j * x_i \vee b_{i,j} * x_i \vee x_j * g_{i,j}$. In this proposal, each particle can accomplish four types of moves. Three involve the action of another particle (*cognitive*) and in the other only the particle is self-involved (*inertial*). Each particle randomly performs just one of these four moves by evolution. This discrete PSO interprets cognitives moves as crossover operations involving the solution of the moving particle and another solution that acts as an attractor. On the other hand, the inertial move is translated into a mutation operation that randomly changes components of the solution. More information and other proposals for PSO can be found in [1].

5 Vehicle Routing Problem with Time Windows

In the Vehicle Routing Problem (VRP) [5], the goal is to determine a least cost route-plan using a fleet of vehicles to serve certain demand from a set of costumers. The VRP can be formally described as follows. Let $G(V, A)$ be a graph where V and A are set of vertices and arcs respectively. The first vertex v_0 is the depot. A fleet of vehicles of identical capacity Q serve certain demand q_i from a number of costumers $V - \{1\}$. Each arc has a cost associated $c_{ij} \geq 0$, such that $i \neq j$. Moreover, a number of constraints are imposed: 1) each costumer $v_i, i > 0$ can be visited only once, 2) all routes must start and end at the depot and 3) the sum of demands in each route cannot exceed the maximum capacity of each delivery vehicle Q.

In the *Vehicle Routing Problem with Time Window* (VRPTW), both the depot and the costumers have a time window (denoted as $[a_i, b_i]$) in which the delivery must occur. While a late arrival is not usually permitted (hard constraint), arriving before the lower time window limit a_i is allowed. It this case, the delivery incurs in a *waiting time* until the customer can be served. The VRPTW also takes into account the actual delivery time. This time is called *service time* and is usually denoted as s_i. A survey with formulations and solution methods for VRP and VRPTW can be found in [10].

6 Experiments

In this section, we describe the settings used in our experiments. In our previous work [2], DLA was compared against Pareto and Lexicographic ranking for selecting the leading particle in the neighborhood of each moving particle. However, this time DLA is presented in two versions for selecting the leaders and also updating the best personal position b_i and the best position achieved by the swarm so far g. The first version of DLA uses a greedier approach to establish the probability for each preference using the *pmf* $p(x) = 0.9 * exp(-x) + 0.05$. The first coefficient (0.9) increases its curvature and the second moves it up by 0.05. We set these coefficients according to some preliminary tests. The second version of the DLA (DLA^2) splits the ranking process into two phases. If the current number of iterations is less or equal than a given k, a probability mass function is used to encourage intensification. Otherwise, a different probability mass function is employed to encourage diversification. To this aim, the first phase only takes into account the first two highest preferences using this *pmf* $p(x) = 0.6 * cos(0.7x + 0.1) + 0.3$ with $x = \{0, 1\}$. For values between $x = 0$ and $x = 1$, this function (equivalent to a quadratic expression in the given range) assigns high and similar probabilities to the two objectives with the highest preference. The coefficients were set, as in the previous *pmf*, using preliminary computational experiments. This intensification phase lasts k iterations. In our experiments k is set to 500 which corresponds to 25% of the total number of iterations that the algorithm runs. The diversification phase runs for the remaining iterations using the function mentioned above $p(x) = 0.9 * exp(-x) + 0.05$, working with the whole set of preferences.

Our efforts focus on comparing different ranking approaches. For this purpose, we implemented a canonical PSO inspired in the Discrete-PSO (DPSO) proposed by Consoli et al. [4]. This approach, as explained in Section 4, allows particles to move in four possible directions depending on which attractor each particle follows. The probability of all attractors were set to 0.25. In our simulations, the swarm was formed by 50 particles evolving for 2000 generations. We used the 100 costumers Solomon's [6] instances of the VRPTW. These instances are divided in three classes: *c1xx* - costumers positioned in clusters, *r1xx* - costumers randomly spread and *rc1xx* - some costumers forming clusters and others randomly positioned. Regarding the DPSO implementation, two operators were used. The crossover operator copies a random route from an attractor to the moving particle. The mutation operator exchanges costumers from one route to another within the route-plan in the solution of the moving particle.

In order to assess the performance of each ranking approach, a number of objectives were considered: *Travel Time* (Z_{tt}) or elapsed time of the route-plan, *Waiting Time* (Z_{wt}) or time the drivers need to wait in case of an early arrival, *Travel Distance* (Z_{td}) or length of the whole route-plan, *Time Window Violation* (Z_{twv}) or sum of lateness of all arrivals, *Number of Time Window Violations* (Z_{ntwv}) or number of customers not served within the appropriate time, *Capacity Violation* (Z_{cv}) or amount of exceeding capacity on vehicles and *Number of Capacity Violations* (Z_{ncv}) or number of vehicles whose capacity is being exceeded. Reducing violations are considered as objectives in this study. In this way, we entitle the decision-maker to decide on the convenience of serving costumers out of their time windows or exceeding the capacity of some vehicles.

In a preliminary study, we tested a number of different combinations of objectives using Pareto dominance. These experiments showed that the best setting for Pareto dominance is to discriminate solutions using (Z_{td}, Z_{twv}). For the Lexicographic approach, a similar study was carried out finding that the best sequence is $(Z_{ntwv}, Z_{td}, Z_{wt}, Z_{tt}, Z_{twv}, Z_{cv}, Z_{ncv})$. Both studies were based on the results of the best extreme values achieved by each combination. To this aim, we took those combinations that made the search process converge faster to feasible regions, analyzing Z_{twv} and Z_{cv}. With respect to the DLA and DLA^2, they both used the same sequence for preferences as the lexicographic for priorities. But, for DLA^2 this sequence was used after 500 generations as explained above.

7 Results

In order to analyse the performance of the proposed DLA variants, we present the results in two modes. Firstly, we depict the approximated non-dominated sets obtained by each strategy using two objectives. Secondly, we show a table containing the normalized average *S-metric* [11] values for each instance family and technique. Three pairs of plots are shown in Figure 1. Each pair shows the approximated non-dominated sets obtained when using Pareto dominance, Lexicographic ordering, DLA and DLA^2 on instances $c101$, $r101$ and $rc101$ using two pairs of objectives. Due to space limitations, for both the plot and the S-metric, we only show the most meaningful combinations of pairs of objectives.

That is, *Time Window Violations* (Z_{twv}) vs *Travel Distance* (Z_{td}) and *Travel Time* (Z_{tt}) vs *Travel Distance* (Z_{td}).

Respect to the Z_{twv} vs Z_{td} comparison, DLA^2 is clearly superior in both intensification and diversification. The approximation sets obtained by DLA^2 on $c101$, $r101$ and $rc101$ have solutions with closer values to the origin, revealing better intensification behavior. Additionally, these solutions, compared to those of other ranking methods, seem to be more spread along both axis, showing better diversification. Moreover, DLA^2 seems to get better results as the difficulty of the instances increases. Analyzing the properties of the Solomon's instances, it is observed that the time windows are much more restrictive on instance sets $r1xx$ and $rc1xx$. Moreover, the geographical location of costumers in these two sets of instances make the problem grow in complexity. This complexity is due to the fact that optimizing the distance does not guarantee to obtain a good set of solutions. This improvement can be seen on the results for instances $r101$ and $rc101$ where the difference in performance between DLA^2 and the other ranking methods is much more noticeable. For this comparison, DLA gets a slightly better performance than the Lexicographic approach. This is because DLA uses a greedy function to assign probabilities to each preference. Therefore, the priority vectors were generated within a short distance with the one used for the Lexicographic ordering. Pareto shows a reasonable performance in instances $r101$ and $rc101$, but it performs poorly on instance $c101$.

In the comparison of Z_{tt} vs Z_{td}, we do not appreciate a good diversification performance in any technique. This is because the Solomon's instances were not created to be used as a multi-objective test case. These instances have symmetrical and identical matrices for distances and times. The variability when comparing these two objectives arises mainly from the waiting and service times. However, this comparison shows the intensification achieved by each ranking technique. DLA^2 is again the best technique obtaining much better results than the others in these three instances. DLA produced approximation sets with more intensification than Lexicographic in these plots as well. Pareto presented a similar behavior to the one exposed for the other objectives comparison, providing in general a poor performance.

Table 1 shows the performance of each ranking approach on each instance set, calculated with the S-metric [11] or hyper-volume. This metric computes

Table 1. Performance of different ranking approaches on Solomon's instances according to the S-metric quality measure calculated over two comparisons: *Time Window Violations* (Z_{twv}) vs *Travel Distance* (Z_{td}) - on the left, and *Travel Time* (Z_{tt}) vs *Travel Distance* (Z_{td}) on the right. Average values and their respective standard deviations are shown for each family of Solomon's instances ($c1xx$, $r1xx$ and $rc1xx$).

	Z_{twv} **vs** Z_{td}				Z_{tt} **vs** Z_{td}			
	Pareto	**Lex**	**DLA**	**DLA^2**	**Pareto**	**Lex**	**DLA**	**DLA^2**
c1xx	.20(.17)	.69(.08)	.53(.18)	1(.0)	.50(.14)	.89(.06)	.74(.21)	1(.0)
r1xx	.21(.08)	.28(.12)	.27(.10)	1(.0)	.72(.13)	.72(.09)	.71(.09)	1(.0)
rc1xx	.11(.04)	.25(.10)	.2(.14)	1(.0)	.57(.07)	.65(.88)	.67(.10)	1(.0)

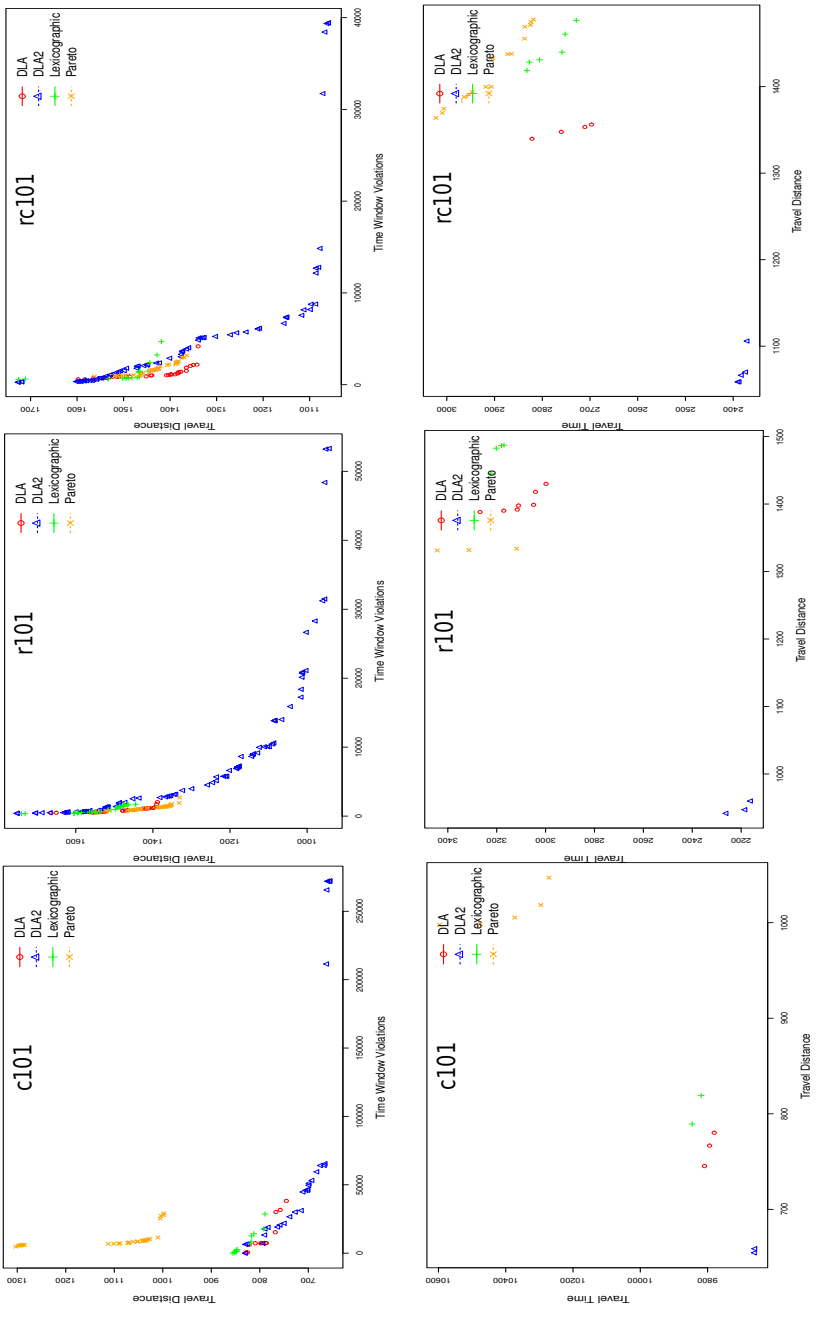

Fig. 1. Approximation sets from different ranking techniques on the Solomon's instances $c101-$ first row, $r101-$ second row and $rc101-$ third row. *Time Window Violations* (Z_{twv}) vs *Travel Distance* (Z_{td}) - on the left, and *Travel Time* (Z_{tt}) vs *Travel Distance* (Z_{td}) on the right.

the hyper-volume of the space limited by the solutions in an approximation set and a reference point. The larger the value of the S-metric, the higher the quality of the approximation set. We calculated the S-metric using the same pairs of comparisons used for the previous graphs. So, these results extend the information conveyed by the plots. All the values were normalized to a value between 0 and 1 according to the highest hyper-volume value obtained in each case. Table 1 is divided in three columns. The first one identifies the instances used. The second and third columns show the S-metric value comparing Z_{twv} against Z_{td} and Z_{tt} against Z_{td} respectively.

For the first comparison, *Time Window Violations* (Z_{twv}) vs *Travel Distance* (Z_{td}), DLA^2 gets the best S-metric value for all instances. On average, it obtains an improvement of 60% over Lexicographic ordering which is the second best ranking method almost in a tie with the DLA. Pareto dominance gives the worst performance with an average of 0.18 across all instances. In the second comparison, *Travel Time* (Z_{tt}) vs *Travel Distance* (Z_{td}), DLA^2 is not the best in only one instance: $c108$. However, it is very close to that value with a distance of only 0.5%. In general, DLA^2 presents an improvement of about 25% over the Lexicographic ordering, which is again the second best ranking technique. Very close to the Lexicographic approach, the DLA gets better results in some instances and worse in others, but on average the former outperforms the later on about 5%. Pareto dominance comes last with an overall performance of 0.60.

The Frieman test was used to analyse the global differences in the S-metric values obtained for each method in both comparisons. There are significant differences at 0.01 significance level. In order to find out where these differences are, we carried out a number of pair-wise comparisons using the four ranking approaches with the Wilcoxon test. According to this test, there is a significant difference between DLA^2 and Lexicographic ordering at 0.01 significance level. The normalized values for the statistic were 4.70 and 4.68, respectively. So, we can safely say that DLA^2 is superior to Lexicographic ordering in this scenario. Similarly, we compared DLA and Lexicographic ordering obtaining the two-way p-values 0.1236 and 0.1285, respectively. This reveals that there is not significant difference between these two ranking approaches at 0.10 significance level. Finally, DLA and Pareto dominance were compared obtaining the two way p-values 0.0007 and 0.0434, respectively. Thus, their results are significantly different at 0.05 significance level.

8 Conclusions

We presented an extended study of a novel approach to rank solutions in multi-objective optimization problems. Our simulations show that Dynamic Lexicographic Approach (DLA) is a valuable technique to discriminate solutions. Specially the variant with double *pmf* (DLA^2) in which the combination of intensification and diversification exhibits much better performance than the other ranking approaches such as Lexicographic ordering and Pareto dominance.

The success of the Dynamic Lexicographic Approach here indicates that it is important to question standard ranking approaches and their suitability for

certain problems. This is specially important in those problems where we deal with a large number of objectives. DLA is an alternative easy to implement and intuitive for the decision-maker.

In future research, we will extend these experiments to other MOO problems such as the multi-objective TSP. We are also developing a set of truly multi-objective VRPTW instances based on real-world problems in which the time and distance matrices are not symmetrical nor identical. Moreover, time windows for costumers will be more relaxed than in many other instances in the literature. Another issue that is worth of an in-depth study is the adaptability of the DLA respect to the probability mass function (pmf), including setting the parameter k to switch between intensification and diversification.

References

1. Banks, A., Vincent, J., Anyakoha, C.: A review of particle swarm optimization. part II: hybridisation, combinatorial, multicriteria and constrained optimization, and indicative applications. Natural Computing 7(1), 109–124 (2008)
2. Castro-Gutierrez, J., Landa-Silva, D., Moreno Perez, J.: Dynamic lexicographic approach for heuristic multi-objective optimization. In: Proceedings of the Workshop on Intelligent Metaheuristics for Logistic Planning (CAEPIA-TTIA 2009) (Seville (Spain)), pp. 153–163 (2009)
3. Coello, C., Lamont, G., Veldhuizen, D.: Evolutionary algorithms for solving Multi-Objective problems. In: Genetic and Evolutionary Computation, Springer, Heidelberg (2007)
4. Consoli, S., Moreno-Pérez, J.A., Darby-Dowman, K., Mladenović, N.: Discrete particle swarm optimization for the minimum labelling steiner tree problem. Natural Computing 9(1), 29–46 (2010)
5. Dantzig, G., Ramser, J.: The truck dispatching problem. Management Science 6(1), 80–91 (1959)
6. Dorronsoro Díaz, B.: VRP web (2009), `http://neo.lcc.uma.es/radi-aeb/WebVRP/`
7. Ehrgott, M., Gandibleux, X.: Multiple criteria optimization: State of the art annotated bibliographic survey. International Series in Operations Research and Management Science, vol. (52) Springer/Kluwer (2002)
8. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: IEEE International Conference on Neural Networks - Proceedings, vol. 4, pp. 1942–1948 (1995)
9. Kokolo, I., Hajime, K., Shigenobu, K.: Failure of pareto-based MOEAs, does non-dominated really mean near to optimal? In: Proceedings of the 2001 Congress on Evolutionary Computation (CEC 2001), pp. 957–962. IEEE press, Los Alamitos (2001)
10. Laporte, G., Golden, B., Grazia, M., Melissa, A.: The vehicle routing problem: Latest advances and new challenges. Operations Research/Computer Science Interfaces Series, vol. 43. Springer, Heidelberg (2008)
11. Zitzler, E., Brockhoff, D., Thiele, L.: The hypervolume indicator revisited: On the design of pareto-compliant indicators via weighted integration. In: Obayashi, S., Deb, K., Poloni, C., Hiroyasu, T., Murata, T. (eds.) EMO 2007. LNCS, vol. 4403, pp. 862–876. Springer, Heidelberg (2007)