The capacitated vehicle routing problem with soft time windows and stochastic travel times

Jorge Oyola[1]

Fecha de recepción: 9 de octubre de 2018
Fecha de aprobación: 14 de diciembre de 2018

Abstract

A full multiobjective approach is employed in this paper to deal with a stochastic multiobjective capacitated vehicle routing problem (CVRP). In this version of the problem, the demand is considered to be deterministic, but the travel times are assumed to be stochastic. A soft time window is tied to every customer and there is a penalty for starting the service outside the time window. Two objectives are minimized, the total length and the time window penalty. The suggested solution method includes a non-dominated sorting genetic algorithm (NSGA) together with a variable neighborhood search (VNS) heuristic. It was tested on instances from the literature and compared to a previous solution approach. The suggested method is able to find solutions that dominate some of the previously best known stochastic multiobjective CVRP solutions.

Keywords: genetic algorithms; heuristic algorithms; multiobjective programming; random processes; vehicle routing.

El problema de enrutamiento de vehículos con ventanas de tiempo suave y tiempos de viaje estocásticos

Resumen

Un enfoque totalmente multiobjetivo es usado en este artículo para estudiar un problema de enrutamiento de vehículos capacitado (CVRP), estocástico y multiobjetivo. En esta versión del problema, la demanda se considera determinística, pero los tiempos de viaje son asumidos como estocásticos. Una ventana de tiempo suave es asociada con cada cliente y hay una penalización por iniciar el servicio por fuera de esta. Dos objetivos son minimizados, la distancia total recorrida y la penalización por no cumplir con la ventana de tiempo. El método de solución propuesto incluye un algoritmo genético con ordenamiento no dominado (NSGA) y una heurística de búsqueda de vecindad variable (VNS). Se probó en problemas de la literatura y se comparó con un enfoque previo de solución. El método propuesto es capaz de encontrar soluciones que dominan algunas de las mejores soluciones conocidas para el CVRP multiobjetivo.

Palabras clave: algoritmos genéticos; algoritmos heurísticos; optimización multiobjetivo; proceso aleatorio; ruteo de vehículos.

[1] Ph. D. Universidad de Córdoba (Montería-Córdoba, Colombia). jorgeoyola@correo.unicordoba.edu.co. ORCID: 0000-0002-6501-7036.

Revista Facultad de Ingeniería (Rev. Fac. Ing.) Vol. 28 (50), pp. 19-33. Enero-Marzo, 2019. Tunja-Boyacá, Colombia. L-ISSN: 0121-1129, e-ISSN: 2357-5328, DOI: https://doi.org/10.19053/01211129.v28.n50.2019.8782

O problema de roteamento de veículos com janelas de tempo suave e tempos de viagem estocásticos

Resumo

Um enfoque totalmente multiobjetivo é usado neste artigo para estudar um problema de roteamento de veículos capacitado (CVRP), estocástico e multiobjetivo. Nesta versão do problema, a demanda considera-se determinística, mas os tempos de viagem são assumidos como estocásticos. Uma janela de tempo suave é associada com cada cliente e tem uma penalização por iniciar o serviço por fora desta. Dois objetivos são minimizados, a distância total recorrida e a penalização por não cumprir com a janela de tempo. O método de solução proposto inclui um algoritmo genético com ordenamento não dominado (NSGA) e uma heurística de busca de vizinhança variável (VNS). Provou-se em problemas da literatura e comparou-se com um enfoque prévio de solução. O método proposto é capaz de encontrar soluções que dominam algumas das melhores soluções conhecidas para o CVRP multiobjetivo.

Palavras chave: algoritmos genéticos; algoritmos heurísticos; optimização multiobjetivo; processo aleatório; roteamento de veículos.

Para citar este artículo:

J. Oyola, "The capacitated vehicle routing problem with soft time windows and stochastic travel times," *Revista Facultad de Ingeniería*, vol. 28 (50), pp. 19-33, Ene. 2019. DOI: https://doi.org/10.19053/01211129.v28.n50.2019.8782.

I. Introduction

The capacitated vehicle routing problem (CVRP) is a well-known problem in transportation proposed in 1959 [1]. It is defined over an undirected graph $G(V, E)$, where $V = v_0, \ldots, v_N$ is a set of vertices and $E = \left(v_i, v_j\right): v_i, v_j \in V, i < j$ is a set of edges. There is a symmetric matrix $C = [c_{ij}]$ that correspond to the travel costs along edge $(v_i v_j)$. Vertex v_0 represents the depot where there is a homogeneous fleet of m vehicles with capacity Q. A set of customers $V \backslash v_0$ with a non-negative known demand d_i must be served. A solution to the CVRP consists of m delivery routes with some specific conditions. Each route must start and end at the depot. Each customer must be visited once by exactly one vehicle. The summation of the demands of the customers in the same route, must be less than or equal to Q. A different approach where the demand corresponds to items that must be collected from the customers leads to an equivalent problem. The classic objective is minimization of total travel costs [2].

The previous definition holds for the deterministic CVRP, however it is expected that in the real world one or more of the elements of the CVRP will be uncertain. Such elements are included in the models in the form of stochastic parameters. This variant of the problem became known as stochastic (capacitated) vehicle routing problem (SVRP or SCVRP) [3, 4].

Delivery reliability defined as the on-time delivery of products and services is a major competitive arena for many companies [5]. The travel time between two customers may be affected by the congestion in the road and other eventualities, such as accidents, making the CVRP with stochastic travel times a relevant problem to study. This paper deals with a stochastic multiobjective vehicle routing problem with soft time windows (TW). The penalty for servicing the customers outside the time windows is included as objective, in addition to the traditional minimization of the total length. The travel times are assumed to be stochastic. This problem is known as the VRP with soft time windows and stochastic travel times (SVRPSTW).

Real-world transportation planning is a multiobjective problem [6]. It can be expected that including additional relevant objectives into the model will make it more realistic. In a multiobjective optimization problem (MOP) several functions are optimized (minimized or maximized) subject to the same set of constraints. As a general rule, no single solution can minimize all objectives simultaneously. Instead of that, a solution to a MOP is given by a set of solutions, which are called tradeoff solutions [7], Pareto optimal solutions or Pareto Set [8].

The tradeoff solutions consist of the set of non-dominated solutions. A solution y with objective function values $\left(f_1(y), f_2(y), \ldots, f_n(y)\right)$, dominates a solution z, $y \prec z$, if and only if, the solution z does not perform better than y in any objective function, but it performs worse in at least one.

When dealing with stochastic MOP the dominance may be evaluated in different ways [9], the expected values of the objective functions are used here to compare. A multiobjective model for the SVRPSTW was formulated in [5]. However, their solution approach was not entirely multiobjective. In contrast, we attempt to approximate the Pareto set.

A. Problem definition

A multiobjective approach to the CVRP with soft time windows and stochastic travel times (SVRPSTW) is found in [5]. In SVRPSTW the demand is known in advance, and there is a deterministic service time and a time window $[e_i l_i]$ associated with each customer i. Starting service outside the time window is allowed at a cost, either for earliness or lateness. The problem is modeled as stochastic programming with recourse (SPR), the recourse being the cost for servicing outside the time windows. The development of the mathematical model to the problem is explained in detail in [5].

Even though the problem is modeled as a multiobjective problem, just two potentially Pareto solutions are found. The three objective functions are combined into a weighted single objective which is later minimized by the tabu search algorithm. Tests were performed using two different sets of weights, one solution is found in every case.

A bi-objective reformulation of the SVRPSTW is presented here, where both the expected total distance and the expected penalty cost (for starting the service outside the time windows) are minimized. The Number of vehicles is given as parameter, in principle it is possible to use the value found in [5]. However, it is possible to set different values to the number of vehicles and solve the bi-objective problem for each value. In that case, the obtained sets of solutions will approximate the Pareto set to the original SVRPSTW.

The obtained solution sets are compared against the solutions presented in [5]. The fact that the solution to the problem is presented in a form of approximation to the Pareto set, becomes one main contribution of this work. Since no previous attempts have been made to approximate its Pareto set. Another contribution is the algorithm that succeeds in finding such approximation.

B. Travel time and closed-form expressions for the penalties

The travel times are considered to follow a shifted gamma (α, β, δ) distribution, where α is restricted to take only integer values. This condition on the travel time allows the exact penalty computation, at the cost of generality. The arrival time at customer i, τ_i, is equal to the cumulative travel times plus the sum of the service times of all preceding customers. The earliness penalty at customer i, Ξ_i, was expressed in [5] as in Equation (1).

$$\Xi_{\mathrm{i}} = \alpha\beta^2 \left[\left[\left(\frac{E_i}{\beta}\right) - \alpha' \right]^2 + \alpha' - \left(\left\{ \alpha' \left[(\alpha'+1) + (\alpha'-1)\left(\frac{E_i}{\beta}\right) \right] + \sum_{r=2}^{\alpha'-1} \frac{r(r-1)\left(\frac{E_i}{\beta}\right)^{\alpha'+1-r}}{(\alpha'+1-r)!} \right\} e^{-\frac{E_i}{\beta}} \right) \right] \quad (1)$$

Where $E_i = e_i - w_i - \delta \sum_{j \in P_{ik}} d_{jj'} - \sum_{j \in P_{ik}} (s_j + w_j)$, $\alpha' = \alpha \sum_{j \in P_{ik}} d_{jj'}$, $\alpha \in \mathbb{Z}^+, \beta \in \mathbb{R}^+$, $\delta \in \mathbb{R}_0^+$, α is a weight coefficient, j' is the customer immediately following j in route k, P_{ik} is the set of customers served before i in route k.

On the other hand the tardiness penalty at customer i, Λ_i, can be expressed as in Equation (2).

$$\Lambda_i = b\beta^2 \left[\alpha' \left[(\alpha' + 1) + (\alpha' - 1)\left(\frac{L_i}{\beta}\right)\right] + \sum_{r=2}^{\alpha'-1} \frac{r(r-1)\left(\frac{L_i}{\beta}\right)^{\alpha'+1-r}}{(\alpha'+1-r)!}\right] e^{-\frac{L_i}{\beta}} \qquad (2)$$

Where $L_i = l_i - w_i - \delta \sum_{j \in P_{ik}} d_{jj'} - \sum_{j \in P_{ik}} (s_j + w_j)$, b is a weight coefficient.

The weight coefficients a in earliness and b in the tardiness penalty are assumed to be equal to one. The penalty for serving the customers outside the time windows is assumed to be a quadratic loss function, in that way, larger deviations will cause progressively larger losses. If the service at customer i starts at time τ_i and $\tau_i < e_i$, the earliness penalty will be equal to $a(e_i - \tau_i)^2$. The lateness penalty would be equal to $b(\tau_i - l_i)^2$, in case that $\tau_i > l_i$. A discussion on other alternatives for penalty functions and their mathematical expressions can be found in [5].

The waiting time before each customer, including the depot, is a decision variable. So once a vehicle arrives to a customer, it can immediately serve the customer or it can wait. In [5] this decision is done in a post processing procedure, since the main algorithm assumes zero waiting time.

C. Literature review

There are several variants of the SVRP. One of such variants is modeled using the times as stochastic parameters. It can either be the travel times, service times or both. In [10] a CVRP with soft time windows and stochastic travel and service times was solved by means of a tabu search algorithm. A variant of the previous problem, without stochastic travel times and no time windows, was presented in [11], a generalized variable neighborhood search (GVNS) was proposed to solve it. In [12] the CVRP with stochastic travel time and simultaneous pick-ups and deliveries was solved using a scatter search heuristic. Other methods have been used to solve a variant of the last problem see [13, 14]. In [15] a VRP with stochastic travel and service time is described. A robust CVRP with deadlines and travel time/demand uncertainty is studied in [16]. For a more detailed summary on different SVRP, see [3, 17, 18].

There is scant literature on multiobjective SVRP, to the best of the author`s knowledge examples can be found in [5, 19, 20, 21]. A multiobjective approach of the CVRP with stochastic demands (CVRPSD) was formulated in [21]. In [20] the CVRPSD is not explicitly presented as a bi-objective problem, but a tradeoff between the total expected cost and the probability of the solution suffering a route failure (reliability) is taken into consideration. An extension of the CVRP including location, allocation and routing under the risk of disruption is introduced in [19]. A multiobjective CVRP with soft time windows and stochastic travel times (SCVRPSTW) is found in [5].

II. Methods

Evolutionary algorithms (EA) have been used for dealing with different types of VRP. In [22] an EA was used for solving a multiobjective CVRP. Memetic algorithms have been used to deal with different versions of the stochastic VRP [23, 24]. In [21] an EA was used to approximate the Pareto set of a multiobjective SVRP. This may be an indication of the potential of the EA for dealing with multiobjective VRPs and SVRPs.

The target aiming Pareto search (TAPaS) algorithm [8] takes an approximation $\mathrm{PS}_{\mathrm{appr}}$ to a Pareto set and improves it by using a tabu search. An algorithm that follows the same principle as TAPaS, is proposed here, where starting from the approximation obtained from a EA, a local search procedure is applied to improve the quality of the solution. An implementation of the NSGA-II [25], using particular characteristics for the CVRP [26] will be used as EA.

A local search that uses different neighborhood structures, as in generalized variable neighborhood search [11] is applied after the EA. Different neighborhoods are used to attempt to optimize the different objectives and will be applied sequentially. Moves leading to solutions that dominate the incumbent solution are accepted. In [11] the best improvement rule is applied, in this case the application of such rule can be complex and expensive, so the first improvement rule is applied here.

A. Construction of initial solutions

A set of initial solutions is required. We propose a simple deterministic construction heuristic. Most likely high quality solutions will not be obtained, but it can be one step ahead of the randomly constructed set. This heuristic is based on the one used in [27], which is a modified version of a classical insertion heuristic proposed in [28]. An initial diverse set of VRP solutions is constructed with priority shifting from TW penalty to total length. This is achieved by solving a sequence of deterministic hard time-windows problems, where the original (soft) time windows are extended by a varying slack parameter while the distance is minimized as a single objective.

B. Evolutionary Algorithm (EA)

The evolutionary algorithm used here is an implementation of NSGA-II [25], where two crossover operators are used, Split [26] and RBX [29]. The mutation operator will be Or-opt as in [8, 22]. The move that dominates all the others is accepted. The comparison is done among the neighbors, without considering the initial solution. If no move dominates all the others, among the non-dominated moves, the selection will be done comparing the normalized values of the objective functions.

A 2-opt procedure is included as local search. Such procedure has been used before in [8, 22], but here it deals with two objective functions and include the domination criteria. Given two possible moves, the first lead to solution y and the second to solution x, if they are non dominated, $\exists\, A \subset \{1,2,\dots,n\}\colon \forall i \in A\; f_i(y) < f_i(x)$, and $\exists\, B \subset \{1,2,\dots,n\}\colon \forall j \in B\; f_j(y) > f_j(x)$, y will be preferred if $\sum_{i\in A}(1 - f_i(y)/f_i(x)) > \sum_{j\in B}\big(f_j(y)/f_j(x) - 1\big)$. If there is no inequality, one solution is selected randomly.

C. Local search

Once the EA has been executed, every found potentially non-dominated solution is subject to a local search procedure, based on the GVNS heuristic in [11]. In the local search just moves that lead to a solution that dominates the incumbent solution are accepted.

The local search (VNS) performs the search using different neighborhood structures. As in [11], there are inter- and intra-route operations. However, the number of neighborhood structures used in the local search is lower, since the computational requirements are higher. For accepting the moves, the first improvement rule is applied. It assumed that a move will improve the incumbent solution if it leads to a neighbor solution that dominates it or to a solution that does not dominate it, but the improvement in one of the objectives is greater than the deterioration in the other one, as percentage of the current values. A random inter-route swap move is used as a shaking operator. The different neighborhoods are: intra-route insertion moves, intra-route swap moves, inter-route swap moves and Or-opt procedure.

D. Post-optimization

The waiting time at every node, including the depot, is a decision variable. Departure from the depot can be postponed as well as the beginning of the service at every customer. In [5] this decision is done in a postprocessing procedure, since the main algorithm assumes zero waiting time at customers, not at the depot. Waiting times at the depot are computed based on the expected travel times to the first customer in the routes. If the expected travel time from the depot to the next customer is shorter than the early TW parameter associated to such customer, the waiting time at the depot will be equal to the difference, zero otherwise. The postprocessing procedure may change that value. The generalized reduced gradient method is used to deal with such variables in [5]. In our approach the decisions regarding waiting times are also part of a postprocessing procedure. We rely on the quasi-Newton Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm [30] in Scilab for doing such procedure, since we find very convenient to access it from C++ and it is an open source package.

E. Solution evaluation

The equations used by [5] to compute the exact value of the penalties and previously described in Section 1.2, require massive computation, since they deal with transcendental functions and operations with large integers. Such operations can eventually lead to an integer overflow. As a mechanism to avoid such condition, natural logarithm transformations are used, which increases the computational requirements. An evaluation strategy is proposed looking to overcome such difficulty. The evaluation of the solutions is done in different ways along the search process, an idea borrowed from [31]. Let us say that the EA has a maximum number of generations GEN. This total number of generations is divided into three parts from zero to gen_1, from gen_1 to gen_2 and from gen_2 to GEN. During the execution of the algorithm, while running within the first part of the generations, the solutions and the moves are evaluated using deterministic values, and the travel time from a customer i to a customer j

is given by the expected value $(\alpha\beta + \delta)d_{ij}$. In the second part of the generations the solutions and moves are evaluated using sample scenarios. And only in the third part of the generations, the exact value for the expected penalties is computed using the closed-form in equations 1 and 2. Lookup tables were implemented for some of the computations, as a mechanism to reduce the processing time.

Using scenarios for evaluating the solutions has an additional advantage in a different context. If the expected values of the objective functions are unknown, this can be approximated with scenario evaluation. The values would not be exact, but the algorithm becomes independent of the probability distributions of the stochastic parameters. This is out of the scope of this work, however, it is an aspect worth of comment.

F. Test environment

The same instances as in [5] are used. These are a modified version of Solomon instances [28]. Four base test problems are used, R101, R102, R103 and R109. Four different sets of parameters for the travel time probability distribution (α, β and δ) are used with each instance, (1.00, 0.25, 0.75), (1.00, 0.50, 0.50), (1.00, 0.75, 0.25) and (1.00, 1.00, 1.00), identified as S1, S2, S3 and S4 respectively. The number of vehicles is also predefined and is set to be the same as in [5], which means that there may be two versions of the same instance, with a different number of vehicles available. As an illustration, we can say that the instance R101-S3-18V, corresponds to the instance R101, where the parameters of the travel time probability distribution are given by the set (1.00, 0.75, 0.25) and there are 18 vehicles available. In total there are 27 instances for computational experiments. Ten different runs per instance were performed.

The parameters of the algorithm were tuned by means of preliminary testing. A subset of eight test instances was used for the tuning. Each instance had 100 customers, but different set of parameter for the travel time probability distribution and different number of vehicles. Different values were tested for the parameters and these leading to better results, in the first 100 generations, were selected. In some cases, as in the number of generations or population size, also the execution time was taken into account. Regarding the number of generations, several values were tested, including 100. Unless stated otherwise, the values given to these parameters are: number of generations for the NSGA, $GEN = 300$; population size for NSGA, 150; probability of applying RBX crossover operator, $prob_{co} = 0.5$; probability of mutation, $prob_{mt} = 0.4$; number of iterations for VNS, 100; randomization factor of construction procedure, $l_f = 5$; number of generations evaluated deterministically, $GEN_1 = 0.5GEN$; number of generations evaluated by scenarios, $GEN_2 - GEN_1 = 0.25GEN$; number of scenarios, 20 (stability tests were carried out [32]); and maximum number of consecutive customers to move in the Or-opt procedure, 3. All computational experiments were conducted on a computer with processor Intel (R) Xeon (R) CPU E31270 @ 3.40 GHz and 16.0 GB of RAM.

III. Results and discussion

The suggested approach (NSGA&VNS) can find solutions that dominate the solutions reported in [5]. In 12 out of 27 instances such solutions are found even without applying the postprocessing procedure. Once the postprocessing procedure is applied, solutions dominating the ones in [5] are found in 18 out of 27 instances, however in one of the instances, R109-S3-12V, just one of the two solutions is dominated. Results are presented in Table 1. It is emphasized that the NSGA&VNS solutions in Table 1 do not correspond to the approximation of the Pareto set, these solutions are only a subset of the solutions that NSGA&VNS is able to find.

Table 1. Solutions after postprocessing that dominate solutions reported in [5].

Instance	Solutions found in [5]		Solutions found by NSGA & VNS	
	Total length	TW penalty	Total length	TW penalty
R101-S1-17V	1 806.24	667.44	1 598.32	653.79
			1 718.13	630.79
			1 786.99	354.26
R101-S1-18V	2 104.32	212.90	1 947.11	208.07
			1 996.94	190.44
			2 046.90	143.67
R101-S2-18V	2 088.24	646.84	1 851.43	639.15
			1 874.96	592.95
			2 006.97	491.64
R101-S3-16V	1 820.28	2 255.89	1 583.34	2 246.78
			1 705.52	1 814.89
			1 782.88	1 650.55
R101-S3-18V	2 164.59	1 910.68	1 717.20	1 859.80
			1 676.35	1 615.93
			1 948.05	1 272.16
R102-S1-16V	1 585.93	225.82	1 520.03	207.48
			1 568.09	158.72
			1 583.70	131.88
R102-S1-17V	1 792.34	161.54	1 557.97	156.34
			1 647.29	108.10
			1 788.42	47.33
R102-S2-15V	1 695.51	977.96	1 387.17	955.11
			1 544.09	476.55
			1 689.35	336.09
R102-S2-17V	1 826.80	308.07	1 690.62	299.84
			1 737.33	284.80
			1 825.78	250.39
R102-S3-16V	1 661.68	1 138.40	1 484.02	1 129.57
			1 522.54	972.05
			1 648.77	764.54
R102-S3-17V	1 871.34	920.66	1 573.11	911.66
			1 596.70	863.65
			1 683.94	743.77
R103-S1-12V	1 465.64	375.41	1 190.59	323.63
			1 254.98	104.59
			1 406.24	55.15
R103-S2-12V	1 398.88	671.46	1 197.69	617.22
			1 208.97	565.55

			1 389.65	211.84
R103-S2-14V	1 466.74	142.02	1 385.06	141.56
			1 450.25	132.55
			1 459.81	122.86
R103-S3-13V	1 372.24	689.17	1 292.44	542.00
	1 462.45	556.55	1 296.76	530.46
			1 394.20	431.25
R109-S1-12V	1 215.22	11.54	1 195.56	5.51
	1 216.62	6.25	1 203.26	4.83
			1 213.38	2.29
R109-S2-12V	1 216.82	42.95	1 215.84	42.49
	1 224.31	28.36		
R109-S3-12V	1 219.31	138.23	1 230.62	93.34
	1 230.82	110.43		

The instance R103-S1-14V is an example of the instances where solutions dominating the solution reported in [5] were not found. Figure 1 shows a section of the Pareto set approximation including both, NSGA&VNS approximation and the solution in [5]. On the other hand, the Figure 2 shows a section of the Pareto set approximation for the instance R101-S2-18V, where it is possible to see that the solution in [5] is dominated by solutions in NSGA&VNS approximation.

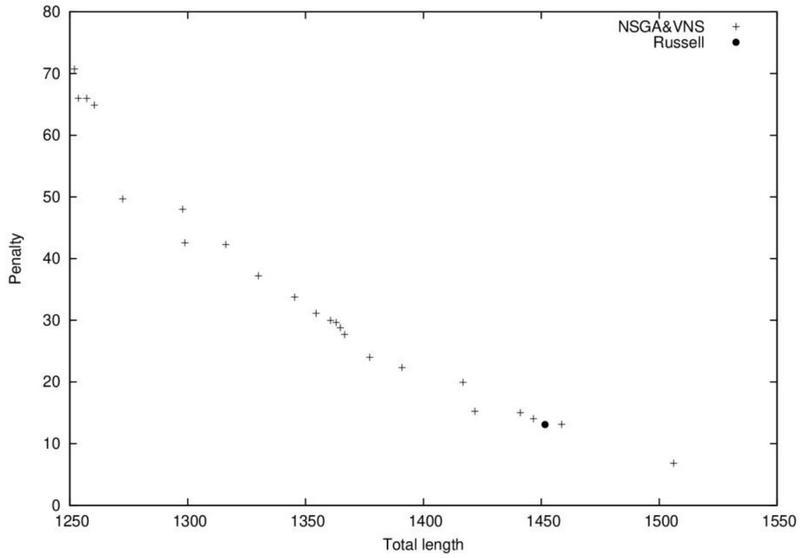

Fig. 1. Our approach finds solutions that do not dominate solution found by [5] in instance R103-S1-14V.

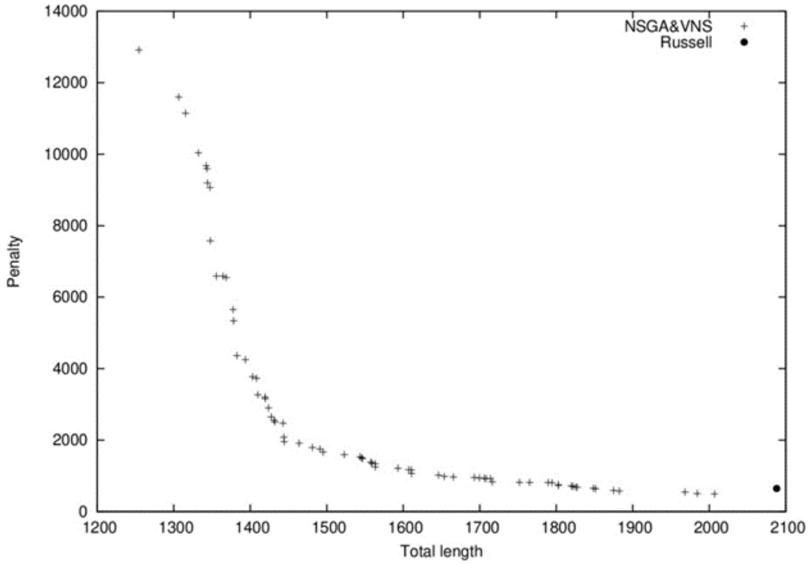

Fig. 2. Our approach finds solutions that dominate solution found by [5] in instance R101-S2-18V.

Table 2. Running time (in minutes).

Instance	Running time in [5]	Total time NSGA&VNS	Instance	Running time in [5]	Total time NSGA&VNS
R101-S1-17V	22.16	2 095.92	R102-S4-16V	351.45	3 280.14
R101-S1-18V	32.03	2 165.31	R102-S4-17V	318.90	3 310.86
R101-S2-17V	87.71	2 935.91	R103-S1-12V	22.43	3 506.49
R101-S2-18V	98.17	3 102.98	R103-S1-14V	48.43	3 922.59
R101-S3-16V	226.78	6 827.92	R103-S2-12V	150.33	4 945.21
R101-S3-18V	245.24	6 776.74	R103-S2-14V	173.30	5 145.21
R101-S4-17V	312.11	2 566.68	R103-S3-13V	657.13*	6 275.71
R101-S4-20V	234.79	2 705.96	R103-S4-12V	398.44	3 014.48
R102-S1-16V	34.05	3 265.88	R103-S4-13V	392.86	3 079.51
R102-S1-17V	40.19	3 377.51	R109-S1-12V	45.43*	2 403.43
R102-S2-15V	114.19	5 201.53	R109-S2-12V	229.06*	3 730.62
R102-S2-17V	127.70	5 691.37	R109-S3-12V	698.97*	4 732.10
R102-S3-16V	261.21	7 872.40	R109-S4-12V	977.02*	2 273.78
R102-S3-17V	257.76	7 601.28			

*The algorithm finds two independent solutions. Time computed by adding both running times.

The running time of NSGA&VNS is much longer than in [5], as it can be seen in Table 2. This can be explained by the fact that while in [5] one solution, at most two, are found per instance, we find an approximation to the Pareto set. As expected, the quality of the Pareto set approximation is improved by the postprocessing procedure. Using the average value of the *S* metric [22, 33, 34] to compare the two approximations to the Pareto set, before and after the postprocessing procedure, it was found that the average improvement over all instances is 0.96%. Russell and Urban [5] reported average improvement above 20%, but such improvement is measured over the objective function value; in contrast, the improvement of the Pareto set is measured here.

A. Impact of NSGA and VNS procedure on the results

A second set of experiments was conducted looking to assess the impact of the NSGA and the VNS procedure in the quality of the obtained solutions. All instances were used in these experiments and ten runs per instance were performed. Three different configurations of the algorithm were compared: the original algorithm described in the former Section (NSGA&VNS) and two algorithms, each consisting in one of the main components of the previous, NSGA and VNS procedure. The running time of the NSGA&VNS was reduced by setting the number of generations for the NSGA component to 30 and the iterations for the VNS to 10. The number of generations of the algorithm consisting of just the NSGA was set to 140, so its running time becomes not shorter than the one used by NSGA&VNS. For the case of the tests using the VNS procedure, the number of iterations was set to 120.

Results were compared using the S and the C metric, as in [22]. It is worth remarking that given two sets of solutions (R,X), the C metric measures the ratio of solutions in X weakly dominated by solutions in R. Using the average value of the metrics, NSGA&VNS performs better than NSGA in 15 out of 27 cases. In six instances is not possible to say which configuration has a better performance since the two metrics are contradictory. On the other hand NSGA&VNS performs better than VNS in 17 out of 27 instances. In eight cases the two metrics lead to contradictory conclusions.

It was observed that VNS is able to find extremal solutions, that do not necessary dominate the solutions found by the other two configurations, but in some cases are good enough to dominate the solutions found in [5]. The NSGA provides a denser set of solutions, which is reflected positively in the S metric. In the NSGA&VNS the solutions found by NSGA are used by VNS afterwards. The latter can improve the solution set by either finding dominating solutions or improving the extremal solutions, outperforming the solutions sets found by NSGA and VNS when executed individually. In conclusion, these experiments indicate that the combination NSGA&VNS works better than each individual method.

IV. Conclusions

The Pareto solutions to a known multiobjective SVRP were approximated for a first time. Obtained results were compared to previously reported individual potentially Pareto solutions. In most of the tested instances, the suggested approach is able to find solutions that dominate the existing solutions, in addition to a wide range of solutions where priority shifts from one objective to the other.

The suggested approach provides good results in most of the tested instances, but still there is room for improvement and several directions can be applied for further research. The efficacy of our approach is decreased when dealing with problems that present a high coefficient of variation. A different construction of initial solutions or local search operators could be adapted to deal with this particular type of instances. A different aspect to consider is the running time, the closed-form expressions for computing the expected value of the TW penalties require a large number of algebraic operations. Perhaps such expressions could be approximated using a method more efficient than the scenarios, but less demanding than the closed-form. Taking waiting times into consideration during the execution of the main algorithm is likely to improve the solutions. This, however, will increase the complexity of the problem. Further work can be done dealing with such complexity and testing the impact on results.

Acknowledgment

Most of the paper was written while the author was at Molde University College (Norway). Their support is gratefully acknowledged.

References

[1] G. B. Dantzig, and J. H. Ramser, "The truck dispatching problem," *Management Science,* vol. 6 (1), pp. 80-91, 1959. DOI: https://doi.org/10.1287/mnsc.6.1.80.

[2] P. Toth, and D. Vigo, "Models, relaxations and exact approaches for the capacitated vehicle routing problem," *Discrete Applied Mathematics*, vol. 123 (1-3), pp. 487–512, 2002. DOI: https://doi.org/10.1016/S0166-218X(01)00351-1.

[3] M. Gendreau, G. Laporte, and R. Seguin, "Stochastic vehicle routing," *European Journal of Operational Research*, vol. 88 (1), pp. 3-12, Jan. 1996. DOI: https://doi.org/10.1016/0377-2217(95)00050-X.

[4] J. Oyola, H. Arntzen, and D. L. Woodruff, "The stochastic vehicle routing problem, a literature review, part I: models," *EURO Journal on Transportation and Logistics*, vol. 7 (3), p. 193–221, Sep. 2018. DOI: https://doi.org/10.1007/s13676-016-0100-5.

[5] R. A. Russell, and T. L. Urban, "Vehicle routing with soft time windows and Erlang travel times," *Journal of the Operational Research Society*, vol. 59 (9), pp. 1220-1228, Sep. 2008. DOI: https://doi.org/10.1057/palgrave.jors.2602465.

[6] J. Current, and H. Min, "Multiobjective design of transportation networks: Taxonomy and annotation," *European Journal of Operational Research*, vol. 26 (2), pp. 187-201, Aug. 1986. DOI: https://doi.org/10.1016/0377-2217(86)90180-3.

[7] Y. Collette, and P. Siarry, *Multiobjective optimization: principles and case studies*, Berlin: Springer, 2003.

[8] N. Jozefowiez, F. Semet, and E. Talbi, "Target aiming pareto search and its application to the vehicle routing problem with route balancing," *Journal of Heuristics*, vol. 13 (5), pp. 455-469, Sep. 2007. DOI: https://doi.org/10.1007/s10732-007-9022-6.

[9] R. Caballero, E. Cerdá, M. D. M. Muñoz, and L. Rey, "Stochastic approach versus multiobjective approach for obtaining efficient solutions in stochastic multiobjective programming problems," *European Journal of Operational Research*, vol. 158 (3), pp. 633-648, Nov. 2004. DOI: https://doi.org/10.1016/S0377-2217(03)00371-0.

[10] X. Li, P. Tian, and S. C. Leung, "Vehicle routing problems with time windows and stochastic travel and service times: Models and algorithm," *International Journal of Production Economics*, vol. 125 (1), pp. 137-145, May. 2010. DOI: https://doi.org/10.1016/j.ijpe.2010.01.013.

[11] H. Lei, G. Laporte, and B. Guo, "A generalized variable neighborhood search heuristic for the capacitated vehicle routing problem with stochastic service times," *TOP*, vol. 20 (1), pp. 99 - 118, Apr. 2012. DOI: https://doi.org/10.1007/s11750-011-0188-6.

[12] T. Zhang, W. Chaovalitwongse, and Y. Zhang, "Scatter search for the stochastic travel-time vehicle routing problem with simultaneous pick-ups and deliveries," *Computers & Operations Research*, vol. 39 (10), pp. 2277-2290, Oct. 2012. DOI: https://doi.org/10.1016/j.cor.2011.11.021.

[13] D. Taş, N. Dellaert, T. Van Woensel, and T. de Kok, "Vehicle routing problem with stochastic travel times including soft time windows and service costs," *Computers & Operations Research*, vol. 40 (1), pp. 214-224, Jan. 2013. DOI: https://doi.org/10.1016/j.cor.2012.06.008.

[14] D. Taş, M. Gendreau, N. Dellaert, T. Van Woensel, and A. de Kok, "Vehicle routing with soft time windows and stochastic travel times: A column generation and branch-and-price solution approach," *European Journal of Operational Research*, vol. 236 (3), pp. 789-799, Aug. 2014. DOI: https://doi.org/10.1016/j.ejor.2013.05.024.

[15] A. S. Kenyon, and D. P. Morton, "Stochastic vehicle routing with random travel times," *Transportation Science*, vol. 37 (1), pp. 69-82, Feb. 2003. DOI: https://doi.org/10.1287/trsc.37.1.69.12820.

[16] C. Lee, K. Lee, and S. Park, "Robust vehicle routing problem with deadlines and travel time/demand uncertainty," *Journal of the Operational Research Society*, vol. 63 (9), pp. 1294-1306, Sep. 2012. DOI: https://doi.org/10.1057/jors.2011.136.

[17] M. Gendreau, O. Jabali, and W. Rei, "Stochastic vehicle routing problems.," in *Vehicle Routing: Problems, Methods, and Applications, Philadelphia, Society for Industrial and Applied Mathematics*, Nov. 2014, pp. 213-239. DOI: https://doi.org/10.1137/1.9781611973594.ch8.

[18] J. Oyola, H. Arntzen, and D. L. Woodruff, "The stochastic vehicle routing problem, a literature review, part II: solution methods," *EURO Journal on Transportation and Logistics*, vol. 6 (4), p. 349–388, Dec. 2017. DOI: https://doi.org/10.1007/s13676-016-0099-7.

[19] A. Ahmadi-Javid, and A. H. Seddighi, "A location-routing problem with disruption risk," *Transportation Research Part E Logistics and Transportation Review*, vol. 53, pp. 63-82, Jul. 2013. DOI: https://doi.org/10.1016/j.tre.2013.02.002.

[20] A. Juan, J. Faulin, S. Grasman, D. Riera, J. Marull, and C. Mendez, "Using safety stocks and simulation to solve the vehicle routing problem with stochastic demands," *Transportation Research Part C Emerging Technologies*, vol. 19 (5), pp. 751-765, Aug. 2011. DOI: https://doi.org/10.1016/j.trc.2010.09.007.

[21] K. Tan, C. Cheong, and C. Goh, "Solving multiobjective vehicle routing problem with stochastic demand via evolutionary computation," *European Journal of Operational Research*, vol. 177 (2), pp. 813-839, Mar. 2007. DOI: https://doi.org/10.1016/j.ejor.2005.12.029.

[22] N. Jozefowiez, F. Semet, and E. Talbi, "An evolutionary algorithm for the vehicle routing problem with route balancing," *European Journal of Operational Research*, vol. 195 (3), pp. 761-769, Jun. 2009. DOI: https://doi.org/10.1016/j.ejor.2007.06.065.

[23] J. E. Mendoza, B. Castanier, C. Gueret, A. L. Medaglia, and N. Velasco, "A memetic algorithm for the multi-compartment vehicle routing problem with stochastic demands," *Computers & Operations Research*, vol. 37 (11), pp. 1886-1898, Nov. 2010. DOI: https://doi.org/10.1016/j.cor.2009.06.015.

[24] K. Sörensen, and M. Sevaux, "A practical approach for robust and flexible vehicle routing using metaheuristics and Monte Carlo sampling," *Journal of Mathematical Modelling and Algorithms*, vol. 8 (4), pp. 387-407, Dec. 2009. DOI: https://doi.org/10.1007/s10852-009-9113-5.

[25] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comp*, vol. 6 (2), pp. 182-197, Apr. 2002. DOI: https://doi.org/10.1109/4235.996017.

[26] C. Prins, "A simple and effective evolutionary algorithm for the vehicle routing problem," *Computers & Operations Research*, vol. 31 (12), pp. 1985-2002, Oct. 2004. DOI: https://doi.org/10.1016/S0305-0548(03)00158-8.

[27] W.-C. Chiang, and R. A. Russell, "A metaheuristic for the vehicle-routeing problem with soft time windows," *Journal of the Operational Research Society*, vol. 55 (12), pp. 1298-1310, Dec. 2004. DOI: https://doi.org/10.1057/palgrave.jors.2601791.

[28] M. M. Solomon, "Algorithms for the vehicle routing and scheduling problems with time window constraints," *Operations Research*, vol. 35 (2), pp. 254-265, 1987. DOI: https://doi.org/10.1287/opre.35.2.254.

[29] J.-Y. Potvin, and S. Bengio, "The vehicle routing problem with time windows part II: Genetic search," *INFORMS Journal on Computing*, vol. 8 (2), pp. 165-172, May. 1996. DOI: https://doi.org/10.1287/ijoc.8.2.165.

[30] P. Venkataraman, *Applied Optimization with MATLAB Programming*, Hoboken: John Wiley & Sons Inc., 2009.

[31] J. Mendoza, L.-M. Rousseau, and J. Villegas, "A hybrid metaheuristic for the vehicle routing problem with stochastic demand and duration constraints," *Journal of Heuristics*, vol. 22 (4), p. 539–566, Aug. 2016. DOI: https://doi.org/10.1007/s10732-015-9281-6.

[32] M. Kaut, and S. W. Wallace, "Evaluation of scenario-generation methods for stochastic programming," *Pacific Journal of Optimization*, vol. 3 (2), pp. 257-271, 2007.

[33] J. D. Knowles, "Local-search and hybrid evolutionary algorithms for Pareto optimization," Ph. D. Dissertation, University of Reading, Reading, 2002.

[34] P. Mateo, and I. Alberto, "A mutation operator based on a pareto ranking for multi-objective evolutionary algorithms," *Journal of Heuristics*, vol. 18 (1), pp. 53-89, Feb. 2012. DOI: https://doi.org/10.1007/s10732-011-9156-4.