Trajectory optimization for waste collection system using multi-objective evolutionary algorithm

Diego A. Prieto-Pedraza (✉ u1802753@unimilitar.edu.co)

Universidad Militar Nueva Granada <https://orcid.org/0000-0001-9765-6484>

Juan C. Guachetá-Alba

Universidad Militar Nueva Granada

Daniela M. Hurtado-Duque

Universidad Militar Nueva Granada

Mauricio Mauledoux

Universidad Militar Nueva Granada

Oscar F. Avilés

Universidad Militar Nueva Granada

Research

Keywords: Waste management, Genetic algorithm, Multi-objective optimization, Multi-agent system, Bargaining

Posted Date: May 5th, 2020

DOI: <https://doi.org/10.21203/rs.3.rs-21299/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Abstract

The control and correct disposal of waste is a global need, promotes recycling and keeps cities clean. In Bogota, the waste collection system has not made any significant changes during the last years, and although it is not an obsolete system, it is a system that can be optimized. The identification of the type of waste and the disposition of the vehicles fleet according to this classification, allows to obtain a more efficient waste collection system. This paper implements a modification on the NSGA-II multiagent optimization algorithm and the Nash Bargaining, to find the best trajectories that the collection vehicles perform on test points in the city of Bogotá, Colombia.

1. Introduction

Currently the collection of waste in Bogotá, is carried out by five different operators divided into localities, Proambiental is responsible for the localities of Usaquén, Chapinero, Santa Fe, San Cristóbal, Usme and Sumapaz; Lime-Limpieza Metropolitana is in charge of the localities of Teusaquillo, Los Mártires, Puente Aranda, Antonio Nariño, Rafael Uribe Uribe, Tunjuelito, Ciudad Bolívar and Bosa; was awarded to Ciudad Limpia the localities of Kennedy and Fontibón; the operator Bogotá Limpia works in the localities of Engativá and Barrios Unidos; Area Limpia D.C operates in the locality of Suba [1]. This division generates that the collection of waste is more efficient in certain areas of the city. It should be noted that the waste collection system does not takes into account the type of waste, however, a modification to the resolution 668 of 2016, states that the waste must be separated using bags of different color [2].

The assignment of tasks is a key issue of the mechanism of cooperation between agents in multi-agent systems. Systems consisting of collaborating mobile agents have been extensively studied and used in a broad spectrum of applications such as environmental sampling, surveillance, coverage, persistent monitoring, homework assignment, and data collection and information gathering. The general topic includes a network of cooperating mobile agents who need to frequently visit points of interest and transfer data, products or people to a base [3]. There are several applications to solve dynamic tasks, such as linear programming, dynamic programming, game theory, the genetic algorithm or the algorithm of auction, where the goal is not only time optimization, also other restrictions are considered, such as fuel consumption and the problem of assigning weapons targets [4].

To solve the problems of tasks distribution in multi-agent systems, different optimization methods such as Ant Colony [5], Insect swarm intelligence [6], insects social [7], negotiation protocols [8], and among other techniques. All focus on optimizing decision-making algorithms in each of the agents that make up the network. Optimization in displacements is a feature that is sought in various fields, this is why the research about trajectory planning has increased considerably and takes into account not only the time of displacement but also external factors, such as weather conditions [9]. Open programs like Google Maps™ or applications such as Waze have allowed users to find a quick solution and simple to move from one point to another within a city.

Path planning techniques have been linked to multi-agent systems to perform task assignments cooperatives, where various points are available through which the system must pass, and for each agent must be planned a trajectory to be the most optimal in distance, the optimization of tasks is implemented through swarms of particles [10]. The auction algorithm refers to the routes of the *D* lite* based agents. *D* lite* is a type of modification of the *A** algorithm and it can find the shortest route in a dynamic situation [4]. These methods are being used in smart cities to improve the quality of life of the inhabitants and reduce the time in various tasks [11]. The formulation of a single objective is extended to reflect the nature of multi-objective problems when there is no objective function to optimize, but many. Therefore, there is no single solution, but a set of solutions. This set of solutions is found through the use of Pareto Optimality Theory [12]. A common simplified assumption is that an agent's preferences are captured by a utility function. This function provides a map of the states of the world or the result of the utility function from the game to a real number [13]. A common problem in multi-agent systems is deciding how to reallocate a set of tasks between a set of agents. This is known as the task assignment problem.

The egalitarian social welfare solution is especially useful in scenarios where there is no agreement that provides all agents the same utility, since it is guaranteed that each problem has an egalitarian social welfare solution. However, the solution itself may in some cases seem very little egalitarian. The egalitarian solution is the agreement that maximizes the sum of the utilities that are shown in Eq. (1).

$$\partial = \arg \min \sum_i u_i(\partial) \quad (1)$$

The utilitarian agreement is, by definition, an optimal Pareto agreement. There may be more than one utilitarian deal in in case of tie. The utilitarian agreement violates the independence of the assumption of cost equity, the utilitarian and egalitarian solutions are reasonable solutions. Nash proposed a solution that does not violate this assumption. The Nash negotiation solution is the treatment that maximizes the product of costs [8], that is shown in Eq. (2).

$$\partial = \arg \max_{\partial'} \prod u_i(\partial') \quad (2)$$

The Nash solution is also efficient on the Pareto front, independent of the units of the utility, symmetric and independent of irrelevant alternatives. In fact, it is the only solution that satisfies these four requirements. This means that if a solution that satisfies these four requirements is wanted, then the only option is the Nash negotiation solution [8].

NSGA-II is a rapid elitist non-mastered genetic classification algorithm for multi-objective optimization which has been proposed by Deb et al. in 2001 [13]. Progress towards the Pareto set is here due to the Pareto classification that divides the population into non-dominated subsets. First: all individuals not

dominated from population are labeled as rank 1; then they are temporarily removed from the population and the process is repeated: not dominated individuals from the rest of the population are given rank 2, and so successively, until the entire population is classified. The diversity conservation technique is based on the agglomeration distance, one of the possible estimates of the density of the solutions belonging to the same subset not dominated.

The selection of NSGA-II is based on tournaments, that is, to choose an individual for reproduction; the T individuals are randomly chosen from the population and compared to each other using the operator of comparison defined above. The winner then becomes a genitor. NSGA-II replacement is deterministic: it consists of merging all parents and children and choosing the best individuals of N in that global population using the same comparison operator [13].

2. Materials And Methods

The problem requires defining different solutions for each collection vehicles, the multiagent system techniques allow solving the multi-objective problem, optimizing the capacity of waste collected by the vehicle, the time that each of these will take to go from one point to another, and the distance they travel. The most optimal trajectory for collection of detected wastes, was calculated using path planning algorithms.

For the algorithm implementation in Bogotá, different collection points in Kennedy and Fontibón locations are chosen. In total are twenty (20) associated points with light poles and trash cans, places where waste is collected daily, taking into account the place of departure and arrival of the collection vehicles. The coordinates of each point were obtained using Google Maps™, through a Python® script, the different combinations between points were made, obtaining an array where each cell is a pair of coordinates expressed in Google Maps™ URLs.

In order to get the distance and time values between each of the coordinates, the import.io page was used. This page allows to automate the export of different data from web pages like Google Maps™; the file with the different URLs was loaded, obtaining a file with the distance and time values between each pair of coordinates, the procedure is observed in Fig. 1. The data obtained from import.io was processed in another Python script to get two arrays for the distance and the time that takes the displacement from one point to another.

For solve the problem, each agent of the multiagent system is one of the collection vehicles. The scenario raised was where are x number of vehicles to collect n quantity of waste types. For each type of waste, there may be a certain number of vehicles and the objective is to find the shortest and the most equal collection route between vehicles. From this, a cost function is created and applied to each agent shown in Eq. (3); in the function the distance and time values obtained using Google Maps™ and Import.io tool was contemplated.

$$\emptyset = \sum d + k \cdot \sum t \quad (3)$$

Where:

d: is the distance between points.

k: is the value constant for time

t: is the time that takes to travel from one point to another.

This cost function is the one that must be minimized through the NSGA-II algorithm, which is shown in Fig. 2. However, crossing and mutation operations are different because the type of problem being solved.

Because the type of problem is discrete, modifications must be made to the NSGA-II algorithm proposed in the theory, implementing the pseudocode presented in Fig. 3 in which its fundamental calculation structure is preserved. However, for this problem it is not possible to do it in this way because the son could contain repeated numbers, which would indicate that the vehicle would cross more than once by the same point and this is incorrect. Due to this, the fraction of the parent vector is selected and a revision is made of which numbers coincided with those existing in the fraction of the parent vector, in order to exchange the positions of the matching numbers and fill the rest of the fraction randomly with the numbers it initially had. In addition, the corresponding mutation of the evolutionary algorithm was raised in three ways:

- Take an agent from a randomly selected individual and exchange the two-point positions of the individual.
- Take a segment of the chain from one of an individual's agents and reverse it.
- Move a segment from one agent to another, for a randomly chosen individual.

According to the procedure, it is started taking into account that there are 20 points including from where the trucks leave and where they should arrive, which are observed in Fig. 4, the starting point called ' Audifarma - El Ensueño ' indicated with blue and ending the tour in ' Rincon de Modelia ' indicated with red. Different individuals are created that are matrices of size number of trucks x number of points where each of them is a solution to the problem. For study purposes, it was decided to use 8 points and 2 vehicles to compare the result of the algorithm with an enumerative procedure in which all possible results would be known.

3. Results And Discussion

For each vehicle of each individual a cost function is obtained, these functions are plotted in a space of n-Dimensions determined by the number of agents. From this graph, the Pareto front is obtained and

each individual is classified according to the front it is on. Figure 5 presents the Pareto fronts in a problem with two agents and 8 target locations, for enumerative algorithm and evolutionary algorithm, which presents its result in generation 30.

This process is carried out for a comparative purpose, in terms of results by both solution techniques, whose more egalitarian individual has the same value for the cost function as the best individual found on the Pareto front of the enumerative process, concluding a good performance of the evolutionary algorithm. It is noted that the performance of the enumerative process is much slower and more complex for problems with a large number of objectives and agents, compared to the genetic algorithm that facilitates the formulation of the problem for the mixed variables used, as well as its expansion over larger data volumes, being real cases that occur in many cities.

Subsequently, the proposed algorithm is used on the problem of three trucks and 20 target collection sites observed in Fig. 3. Figure 6 shows the Pareto front of the evolutionary algorithm in the 200 generation for the problem, in addition to the cost values obtained by each of them, visualizing a correct distribution of them in comparison to other individuals. It is opportune to show the distributions in two dimensions of the Pareto front present in the solution found, graphs presented in Fig. 7, noting that a lot of solutions present in the problem turn out to be optimal, but selfish about some agents.

It should be emphasized that the pursuit of minimizing cost functions in multi-objectives can result in selfish agents fulfilling all the tasks expressed and be an optimal solution; given this, Nash's bargaining is applied on the optimal Pareto front, a criterion that selects the most efficient individual to solve the given tasks. Due to the nature of the algorithm, individuals in the final generations tend to Pareto front, where all solutions are optimal, as seen in red in Fig. 7. Nash bargaining allows us to select the most egalitarian solution on the recollection problem, which is shown in yellow in Fig. 7; this criterion is fundamental to carry out the search for cost minimization of each objective. This algorithm was implemented using the computing specifications presented in Table 1.

Table 1
Project computing specifications

MATLAB® version	Processor	RAM (GB)	Time running (s)
R2018b	Intel® core™ i7-6700HQ	8	41 – 54

By obtaining the best set of trajectories that agents will perform, we proceed to reconstruct them on the points located on the map. These are seen in Fig. 8, where in turn the distances and times that each of the agents will take to fulfill the trajectory, presenting a correct distribution in the values and on the tasks, they perform are reported.

4. Conclusions

The generation and processing of data taken from Google Maps was carried out in a single instant of time, as this way the times and distances to be performed by the collecting vehicles are correctly captured, taking into account traffic and situations of the road. This formulated algorithm is applicable to a problem of a number of waste collection points, performed by m collection vehicles; the amount shown was selected for a correct visualization and is located in two Bogotá city localities managed by "Ciudad Limpia" company.

Given the nature of the problem, it was necessary to adjust the crossover and mutation methods of the NSGA-II algorithm, because the values for the individuals of the evolutionary algorithm are discrete, it was necessary to perform all programming without relying on apps or Toolbox.

Using the modified NSGA-II algorithm, optimal and egalitarian solutions were found to the multi-agent system in which there are different cases of two or more trucks that must pass through different waste collection points, thus obtaining routes where the trajectory of each truck and the time spent are minimized. In most cases during different tests using about 200 generations, similar results are obtained where the generation of individuals converges to those who are egalitarian.

Declarations

Acknowledgements

The authors would like to thank the Davinci Research Group, Universidad Militar Nueva Granada, Colombia, and its Vice-Rector of Research, for making all the necessary resources available for this work.

Authors' contributions

All the authors proposed the study, defined the structure of the algorithm and participated in the writing of the manuscript. DP performed the programming of the genetic algorithm. JG participated in the design of the study and design the results report. DM carried out the collection and exploration of the data and edited the manuscript. MM and OA conceived the study, defined the method and participated in its coordination. All the authors read and approved the final manuscript.

Funding

The research was funded by the Universidad Militar Nueva Granada, Colombia, and its Vice-Rector of Research, as part of the scientific initiation project called "Garbage classification and trajectory optimization system in waste collection from a simulated plant" PIC-ING-3078.

Availability of data and materials

All data analyzed during this study are recorded from web data exploration.

Competing interests

The authors declare that they have no competing interests.

References

1. A. M. Cuevas, "Estos son los nuevos operadores de recolección de residuos en Bogotá", 2018, <<https://bogota.gov.co/>>.
2. Ministerio de Ambiente y Desarrollo Sostenible, "Minambiente cambia uso de bolsas en Colombia para promover reciclaje y separación en la fuente", 2017, <<https://www.minambiente.gov.co/>>.
3. Khazaeni Y, Cassandras CG, "Event-Driven Trajectory Optimization for Data Harvesting in Multiagent Systems," in IEEE Transactions on Control of Network Systems, vol. 5, no. 3, pp. 1335–1348, Sept. 2018.
4. Kim W, Cho DS, Kim HJ. "Sequential multi-agent task assignment using auction algorithm based on D $\times$ lite," ICCAS 2010, Gyeonggi-do, 2010, pp. 938–942.
5. Wang L, Wang Z, Hu S, Liu L. "Ant Colony Optimization for task allocation in Multi-Agent Systems," in China Communications, vol. 10, no. 3, pp. 125–132, March 2013.
6. Biswas S, Anavatti SG, Garratt MA, "Particle swarm optimization based co-operative task assignment and path planning for multi-agent system," 2017 IEEE Symposium Series on Computational Intelligence (SSCI), Honolulu, HI, 2017, pp. 1–6.
7. Du J, Zhou L, Qu P, Shi Z, Yang L, "Task allocation in multi-agent systems with swarm intelligence of social insects," 2010 Sixth International Conference on Natural Computation, Yantai, 2010, pp. 4322–4326.
8. Luo H, Hu X, Hu X, "Multi agent negotiation model for distributed task allocation," 2010 2nd IEEE International Conference on Information Management and Engineering, Chengdu, 2010, pp. 54–57.
9. IEEE 4th International Conference on Data Science and
Zhao L, Ahmed A, Tang X, Lin N, Liu C, Li J, "A Weather-Assisted Driver Experiences Based Path Selection Method," 2018 IEEE 20th International Conference on High Performance Computing and Communications; IEEE 16th International Conference on Smart City, IEEE 4th International Conference on Data Science and Systems (HPCC/SmartCity/DSS), Exeter, United Kingdom, 2018, pp. 341–345.
10. Biswas S, Anavatti SG, Garratt MA, "Particle swarm optimization based co-operative task assignment and path planning for multi-agent system," 2017 IEEE Symposium Series on Computational Intelligence (SSCI), Honolulu, HI, 2017, pp. 1–6.
11. Nha VTranN, Djahel S, Murphy J, "A comparative study of vehicles' routing algorithms for route planning in smart cities," 2012 First International Workshop on Vehicular Traffic Management for Smart Cities (VTM), Dublin, 2012, pp. 1–6.
12. Ehrgott M. Multicriteria Optimization. Springer, Berlin, second edition, 2005. ISBN 3-540-21398-8.
13. Vidal J. Fundamentals of Multiagent Systems with NetLogo Examples, August 24, 2007.

Figures

Figure 1

Online Tool Import.io a) Interface b) Results

Figure 2

NSGA-II algorithm graphic scheme.

Procedure NSGA-II Modified	Cross Function
Input: np points at map to solve $f_k(X)$ by t agents in g generations Input: n (population size) Initialize population P Q; Generate random population - size n; Evaluate Cost Function; Generate population P -> concerning parents Generate population Q -> concerning children for i=1 to g do for each individual in population do Assign rank based on Pareto Determinate crowding distance end Determinate the new P = best n/2 individuals based on rank and crowding for j=1 to n/2 /each children parentA=Tournment winner based on rank and crowding by rand parents parentB=Tournment winner based on rank and crowding by rand parents new children =cross(parentA,parentB) add new children to new Q end for m=1 to number of individuals to mutate individual=rand() position from P Points=rand(np,2) mutated=mutate(individual, Points) change mutated in new Q end new P= [P Q] end	randpoint1=rand(np) randpoint2=rand(np) choose segments from parentA and parentB based on randpoint1 and randpoint2, find the common numbers in the segments and interchange its positions, fill the remaining positions with the numbers of the original segment.
	Mutation Function
	if randValue()==1 agent=rand(t) interchange two values from individual's agent based on randpoint1 and randpoint2 end if randValue()==2 agent=rand(t) choose a segment from individual's agent based on randpoint1 and randpoint2 and flip it. end if randValue()==3 agent1=rand(t) agent2=rand(t) choose a segment from individual's agent1 based on randpoint1 and randpoint2 and move it to agent2. end

Figure 3

NSGA-II Modified pseudocode

Figure 4

Recollection problem

Figure 5

Pareto front and Nash bargaining for two trucks and 8 points a) Enumerative b) Evolutionary

Figure 6

Pareto front for three trucks and 20 points, generation 200.

Figure 7

Two-dimensional Pareto front, generation 200

Figure 8

Waste collection vehicle trajectories