Using Genetic Algorithm for Advanced Municipal Waste Collection in Smart City

Radek Fujdiak, Pavel Masek, Petr Mlynek, Jiri Misurec
Brno University of Technology
Brno, Czech Republic
Email: {fujdiak, masekpavel, mlynek, misurec}@feec.vutbr.cz

Ekaterina Olshannikova
Tampere University of Technology
Tampere, Finland
Email: {ekaterina.olshannikova}@tut.fi

***Abstract*—The Internet of Things (IoT), as expected infrastructure for envisioned concept of Smart City, brings new possibilities for the city management. IoT vision introduces promising and economical solutions for massive data collection and its analysis which can be applied in many domains and so make them operating more efficiently. In this paper, we are discussing one of the most challenging issues - municipal waste-collection within the Smart City. To optimize the logistic procedure of waste collection, we use own genetic algorithm implementation. The presented solution provides calculation of more efficient garbage-truck routes. As an output, we provide a set of simulations focused on mentioned area. All our algorithms are implemented within the integrated simulation framework which is developed as an open source solution with respect to future modifications.**

I. Introduction

The Smart City represents nowadays hot topic in terms of improving living conditions. Considering mainly the situation in European Union, the EU national governments and also private companies are investing every year significant amount of their budgets to research, development and implementation of the concept of Smart City. Therefore, the term Smart City has many different ways how to define it. In current research, this term is considered as question of ″How to improve a city on different levels?″. Those levels could be related to different stakeholders (i.e. government, authorities, private companies, citizens, etc.) and/or various fields (i.e. mobility, open data, energy efficiency & low carbon solutions, policy & regulation, waste management). The data analysis is a common base-ground for above mentioned issues of the Smart City concept.

The Internet of Things (IoT) is currently considered as a basic communication infrastructure for smart cities, where machines communicate automatically between each other (Machine-to-Machine, M2M) [1]. The biggest advantage is the cooperation of many different communication technologies and devices (machines) within one functional system, where big amount of information and data are shared and used in a secure and smart way [2], [3].

With the Low-Power Wide Area Network (LPWAN), see Fig. 1, IoT idea comes more closer to the real implementation. The LPWAN technologies bring low-cost (1 $ yearly per spot) and low-power (50 µW for connection hub / modem) solutions for million-spot connectivity of various devices [4].

Our work focuses on the optimization algorithms for Smart City management and more specifically this paper deals with municipal waste collection procedure. We consider existing IoT infrastructure and sensor networks, already deployed i.e. in France, where the communication is based on SigFox technology [5]. The new point of view of global IoT infrastructures gives us the possibilities to collect data and, further, deals with common management issues more effectively. Nowadays, the garbage-truck needs to pick-up all garbage cans even when they are empty. Following this fact, we show the way how to use genetic algorithms (GA) as a tool for garbage-collection optimization. The GA should help to use the garbage truck more effectively, i.e. more often in overloaded places. We provide experimental scenarios of different GA use-cases in our simulation environment.

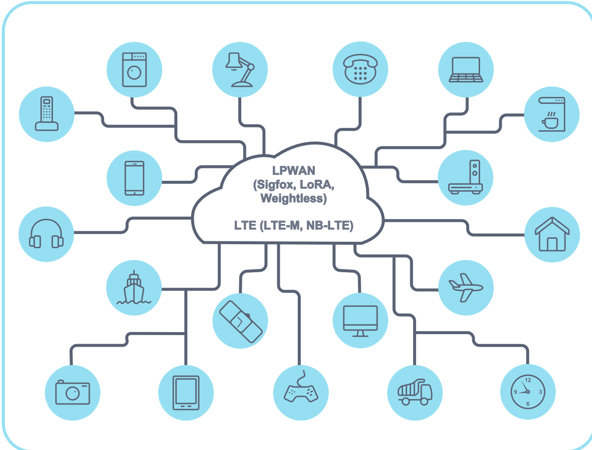

Fig. 1. Low-Power Wide-Area Network (LPWAN) schematic

The rest of the paper is organized as follows. Chapter II is focused on related works in the field of GA utilization for waste collection optimization. We provide summary based on up-to-date works in the field followed by current problems and challenges. Next, chapter III explains our implementation of GA and whole issue of waste collection simulation. Used methods and algorithms are clearly discussed to provide better insight into presented results. Chapter IV discusses the possibility of using the GA in real scenarios for the waste collection issue, comparing the results with current works and giving a hint for real implementation. Chapter VI stands for the conclusion, where our progress and results in the waste collection optimization are summarized together with our future steps.

978-1-5090-2526-8/16/$31.00 ©2016 IEEE

II. Recent Research in Municipal Waste Collection Optimization

The annual and constant growth of population in urban areas brings increasing municipal solid waste generation with socio-economic and environmental impact. Municipal solid waste management - source separation, storage, collection, transfer and transportation, processing and recovery, and, last but not least, disposal, are today current city challenges ([6]–[8]).

Nowadays, advanced technologies and rapid development of Smart City concept allow us to make municipal waste management more intelligent. There is a variety of sensors that could assist different phases of garbage collection. For instance, for the communication we could use weight, filling or position sensors or measuring devices ([9]–[12]). These sensors should be connected to the main infrastructure built in the city, that they can provide the valuable information set for the waste management. In Fig. 2, we can see an example of such considered scenario.

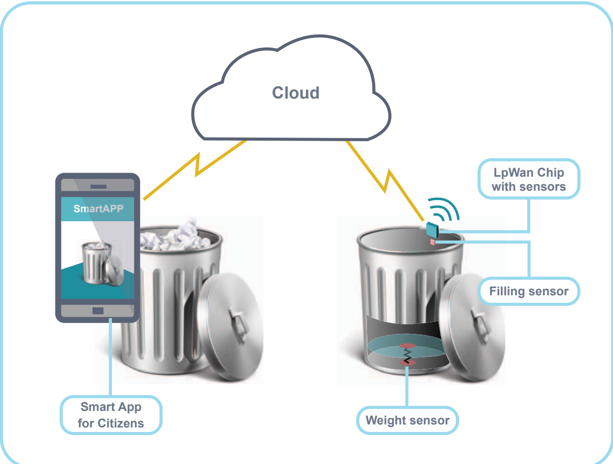

Fig. 2. Considered scenario for municipal waste management

This network could use the LPWAN technologies such SigFox, LoRaWAN, Weightless, LinkLabs, Nwave or different technologies which create low-power and low-cost communication infrastructure [13]. Currently, the main issue regarding waste collection is that the garbage-trucks are operated ineffectively. The cities contain different and various areas, where some bins need to be picked-up more often than others (see Fig. 3).

When we consider installed bins as points without actual information (i.e. about the bin weight), we will lose time, money and also the trucks, which could be used in different places. The sensor network and IoT infrastructure represents a tool for extracting the information from the bins. Further, obtained data might be used for truck-road-calculation and optimization and also for analytical statistics of bin loading (i.e. if the bin will not be overloaded till next collection).

The mathematical programming and processes have been already used for optimizing the municipal waste management and transfer system ([14]–[19]). The waste collection and garbage-truck allocation problem could be solved by traditional mathematical methods such a linear methods ([20], [21]). However, the linear methods show

Fig. 3. Common situation of garbage cans' filling status in todays' cities

insufficient efficiency in some more difficult cases of waste collection. The large amount of variables was the reason for large computation time. The recent research works [22]–[24] use mostly the heuristic solutions and methods dealing with the municipal waste collection as with a Travelling Salesman Problem (TSP). Dealing with problem formulation, the effectiveness of optimization and computation is based on input parameters and specific problem implementation.

Only few works tried to use evolutionary algorithm to deal with implementation and optimization of waste collection problem as the TSP defines. These works [25], [26] use Ant Colony algorithm. However, the genetic algorithm was also proven as a very effective tool to deal with TSP of various implementations ([27]–[30]), but not in the specific implementation of waste collection. We provide in this article experimental measurements focused on the usability of genetic algorithm in the real implementation of the municipal waste collection problem.

III. Implementation of Municipal Waste Collection

The developed software framework represents our implementation made in line with C++11/ISO standard. The Scalable Vector Graphics (SVG files) with map coordinates are used as a program input, which allows easy Extensible Markup Language (XML) editing. We suggest the map as a classical graph representation. The graph G is a pair of parameters (V, E), where V represents vertices (nodes) and E are edges (lines). The E has following dependence:

$$E \subseteq \{\{u, v\} \| u, v \in V\}. \tag{1}$$

Lets have an edge $e \in E$, where e is pair of vertices (u, v) for $u, v \in V$. Then we can say, that the vertices (u, v) are connected by edges e. In our representation, the vertices are the garbage cans and edges the roads. Each road has two parameters for both directions, which represent different weight of each direction i.e. one direction of the road is more loaded than other. The example of this representation is shown in Fig. 4.

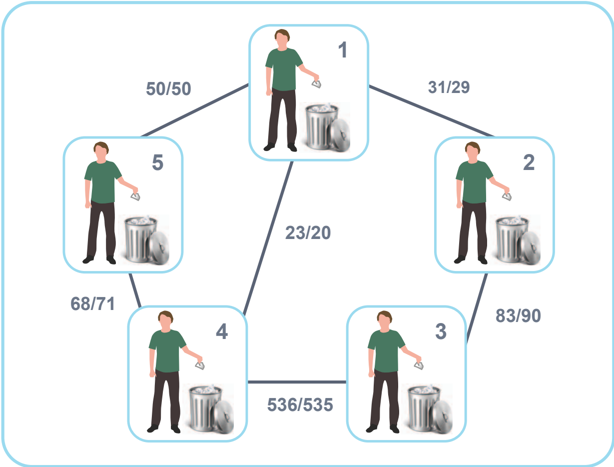

Fig. 4. Example of map in SVG input file

If needed, one-way road might be done with infinite value of the weight parameter. The weight is computed as a sum of all parameters which impact the road (i.e. distance, traffic-density, road quality). The weight of road could be also used as a index of expenses for the road (longer road with high traffic-density will be more expensive than shorter road with fluent traffic). The SVG file consists the basic parameters and attributes for graph creation (i.e. ID of vertex - where it begins and ends, final number of vertices). The basic schematic of our algorithm is introduced in Fig. 5.

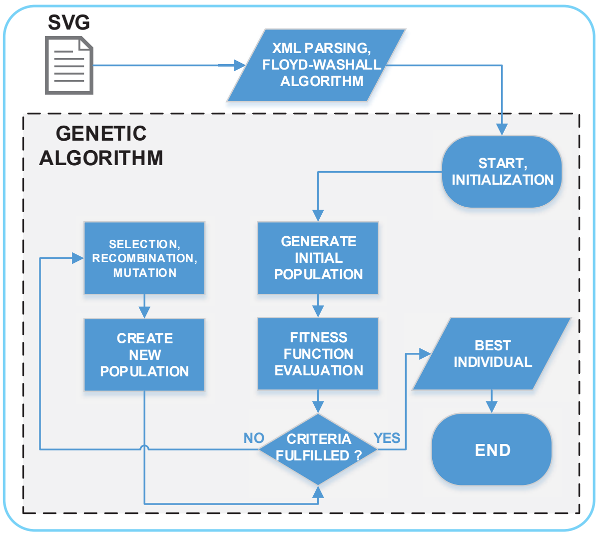

Fig. 5. Functionality of our developed algorithm for optimization

A. XML Parsing and Floyd-Warshall algorithm

The XML parsing is used for the graph (SVG) processing. The main part of the algorithm is depicted in Code 1.

Code 1. XML parsing

```
INPUT: SVG File with graph map
OUTPUT: Graph coordinates, Weight road

From SVG → load distance attribute dist;
create vector[x][y];
vector [x][y]= split(dist);
load end of road → n
x → consists direction of road nodes in order 0 to n;
y → consists direction of road nodes in order n to 0;
return dist, vector
```

The attribute $dist$ consists of the weight of road (vertices) and it is divided by function $split$ into two numbers (two directions) - x, y. This is done for all vertices and saved into 2D-data matrix. After XML parsing, the Floyd-Warshall algorithm is applied to distance recalculation. This algorithm was chosen due to the fact that we are using metric system and there the negative values of edges are not used. The algorithm (Floyd-Warshall) also computes straight the vertices distance, which is less time consuming than i.e. Dijkstr Algorithm (which computes distances always for each vertex). Pseudo-code of used Floyd-Washall algorithm is shown in Code 2.

Code 2. Used Floyd-Warshall algorithm for graph processing

```
INPUT: The adjacency matrix of digraph with n vertices,
       where distances d are ranging from 0 to ∞
OUTPUT: The transitive closure of the digraph

for each vertex v
  d[v][v] ← 0
for each edge (e, v)
  d[u][v] ← w(u,v)
for k ← 1 to n do
 for i ← 1 to n do
  for j ← 1 to n do
   if d[i][j] > d[i][k] + d[k][j]
    d[i][j] ← d[i][k] + d[k][j]
   end if
```

The problem of searching the best way in the graph theory might be represented by the TSP formulation [31]. The graph $G = (V, A)$, where V is vertex $V = \{v_1, ..., v_n\}$ and A is edge with following dependence:

$$A \subseteq \{\{v_i, v_j\} \| v_i, v_j \in V, i \neq j\}. \tag{2}$$

The A is edge with non-negative values of matrix $C = (c_{ij})$ connected with A. In our case, where the roads might have only one direction or the directions might be not same, we must consider the asymmetric TSP, where a condition $c_{ij} \neq c_{ji}$ must be taken into the account. As mentioned, we chose for dealing with the TSP (waste collection) one of the evolutionary algorithm, the Genetic Algorithm. We consider the genetic algorithm as a strong computational and optimization tool, which could be used for more efficient municipal waste management.

B. Description of used genetic algorithm

Our implementation of genetic algorithm is divided into the following basic parts:

- Representation and generation process for population.
- Optimization process of selection, recombination (crossover) and mutation.
- Main criteria for choosing best individual and best solution.

The initial population is given by final number of nodes and first road. The first road is randomly generated. This is also the input parameter for the fitness function. When we have the initial population, the main process of optimization starts.

The selection goes first – we use the truncation selection. It retains the fittest $x\,\%$ of the population. These fittest individuals are duplicated so the population size is maintained. This type of selection has then only one parameter $\tau, \tau \in (0, 1)$, which gives $\tau \cdot N$ best individuals from the old population.

Next step is the crossover and mutation. The crossover is given by a maximum limit of crossing (which is also the minimum value of mutation). From this value the ratio of crossing and mutation is also coming. The dynamical value of crossing given by maximum crossing value is chosen because the same roads can occur in the new population. The same individuals should not be crossed, but mutated to get better or new individual. The crossing algorithm is mentioned in the following code:

Code 3. Used crossover algorithm

INPUT: Random index n, where it will occur the crossing
OUTPUT: The crossed individual (road) p_c

Choose two roads from population p_1, p_2, where $p_1[n] = p_2[n]$
Create empty road p_c
Copy to beginning p_c a part of road p_1: $p_1\{0..n\}$
Copy to end of p_c a part of road p_2: $p_2\{n..\text{end}\}$
Compute the final weight of p_c

The chosen road p_1, p_2 must be appropriated for crossing ($p_1 \neq p_2$). New individual p_c is defined by the group of nodes (road) and road weight. The used mutation algorithm is following:

Code 4. Used Mutation algorithm

INPUT: Random indexes i, j
OUTPUT: The mutated individual (road) p_m

Choose one road from population p_1
Create empty road p_m
p_m[i] = p_1[j]
p_m[j] = p_1[i]
Compute the final weight of p_m

The mutated new individual p_m is again defined by group of nodes (road) and road weight. The final end-parameter for the whole process of the optimization is given by minimum weight change of road over defined time (iterations). This means that if the new solution is not better in α iterations compared with old solution more than $\beta\,\%$, the program will end. This condition secures the algorithm against stacking in local maximum or minimum.

IV. Simulation and Experimental Pre-sets

The simulations and our measurements were done on the machine with following configuration: processor Core 2 Duo E8200 on 2.66 GHz, 4 GB RAM on 800 MHz and 64-bit system. The initial parameters, which are changing in the course of the measurement method, are following: population size 5000, max. crossing 40 %, selection 60 % (throwing 40 %) and 80 vertices.

A. Population Size Impact

The size of population impacts the power requirement, time requirement of the algorithm, but also influences the success rate of finding best solution. This chapter provides measurements analysis regarding the best settings for the population size. The main results are shown in Tab. I.

TABLE I
Impact of the Population Size on the Road (Edge) Weight

Population	Weight span	∅ Weight	∅ Time [s]	Iterations
10	3645–4525	4065	1.3	1683
50	3025–3875	3456	2.2	1694
100	3020–3800	3375	3.4	1352
250	2935–3625	3259	6.1	902
500	2900–3435	3185	10.5	684
1000	2825–3420	3091	18.8	484
2000	2575–3305	2990	34.0	343
5000	2415–3025	2720	92.0	418
10000	2415–2970	2680	160.0	268
15000	2440–2730	2583	263.0	334

The results show that the size of population impacts the success rate of finding the best solution, where the weight of founded best road is linearly decreasing, but it is also exponentially growing the time needed for the computation. The interdependency of road weight and time requirements on the population size is displayed in Fig. 6.

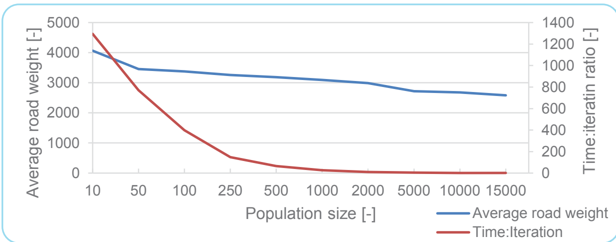

Fig. 6. The interdependency of road weight and time requirements on the population size

As shown, improvement of road weight is more than 25 %, but we need to consider the time:iteration ratio, which exponentially decreases to zero. Around 50 population we achieve 12.5 % of road weight improvement and only 40 % speed decreased. This means that we achieve 50 % of possible improvement with less than 50 % speed decrease. This represents important output, because when the GA will be implemented in the low-power devices, it is necessary to deal precisely with algorithm efficiency and time demands. Smaller effective population (in our case 50 population) should be chosen for low-power solution and if we have *no-power-constraint* implementation then the biggest possible population should be chosen.

B. Population Throwing Impact

The throwing process is connected with selection. We select the best individuals and throw the others for generating new individuals from mutation and crossing. The results of measured throwing impact are provided in Tab. II.

The computation time is oscillating between 60 to 80 seconds. But the time:iteration ratio is decreasing. The interdependency of population throwing, weight and *time:iteration* ratio is displayed in Fig. 7.

Further, we can see the that till 40 % is worth to throw the old population, because it has positive impact on the road weight. More than 40 % of throwing has negative

TABLE II
IMPACT OF POPULATION THROWING ON ROAD (EDGE) WEIGHT

Throwing	Weight span	∅ Weight	∅ Time [s]	Iterations
10 %	2745–2995	2900	70	1945
20 %	2585–3035	2840	68	1296
30 %	2535–2830	2700	62	855
40 %	2480–2960	2640	63	672
50 %	2460–2905	2660	63	486
60 %	2605–2800	2730	75	471
70 %	2550–3085	2770	72	345
80 %	2600–3055	2915	83	257
90 %	2700–3190	2980	72	170

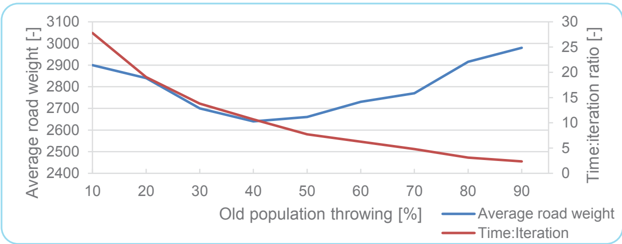

Fig. 7. The interdependency of road weight and number of iterations on population throwing ratio

impact on the road weight with also continuously growing time requirements for one iteration.

C. Crossing and Mutation Ratio Impact

The crossing and mutation are the main methods for optimization in our used genetic algorithm. The crossing creates bigger changes in the new population or individual and it has potential to find better solution as it was found previously. The mutation is doing only small changes with exchanging two nodes and it also helps to the algorithm not to stuck in local minimum or maximum. Then the crossover takes place at the beginning of the optimization process and the mutation at its end. Mutation has also less power requirements and it is possible to handle more mutations in one time unit as crossovers. The main results of measurement of *crossover:mutation* ratio impact are displayed in Tab. III.

TABLE III
IMPACT OF CROSSING:MUTATION RATIO ON ROAD (EDGE) WEIGHT

Crossover	Weight span	∅ Weight	∅ Time [s]	Iterations
0 %	2820–3115	2960	44	1057
10 %	2580–3140	2840	46	574
20 %	2600–2945	2740	54	443
30 %	2560–3050	2750	63	411
40 %	2545–2960	2730	77	381
50 %	2470–3025	2720	92	418
60 %	2440–3020	2740	117	493
70 %	2510–3040	2750	124	425
80 %	2540–2955	2760	129	356
90 %	2665–2875	2770	122	306
100 %	2775–2970	2850	125	216

The results show that it is not possible to deal with the optimization process only with crossover or only with mutation separately. It is also necessary to chose the efficient ratio between these two operations, otherwise the algorithm will be ineffective. The Fig. 8 shows the interdependency of road weight, time to iteration ratio and maximum crossover ratio.

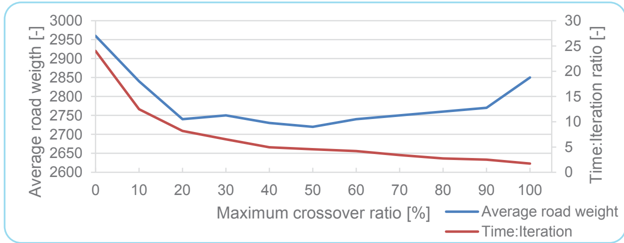

Fig. 8. The interdependency of road weight, time:iteration on the maximum crossover ratio

The biggest change of road weight happening from 0 % to 20 %. More than 20 % decrease exponentially the time:iteration ratio and only very little bit positively impact the road weight. From 50 %, the increasing maximum crossover has negative impact on the road weight. The best ratio from measurements is min. 80 % mutation and max. 20 % crossover setting.

D. Costs saving in the Waste Collection

These measurements compare three different scenarios with 40, 60 and 80 nodes (vertices). With this model, we can estimate the cost saving for one cycle of waste collection. The Tab. IV shows that the average improvement is about 15 % (from 9.52 % to 20.66 %, if we look to the min., max. and average road weight). If we consider the existing IoT infrastructure, the 15 % costs saving is worthy and it is general cost saving for one travelled km by the garbage truck. These resources might be invested i.e. to spread the garbage trucks into the bigger area or more often waste collection.

TABLE IV
COSTS SAVING OF ADVANCED MUNICIPAL WASTE COLLECTION

Vertices	Weight span	∅ Weight	∅ Time [s]	Iterations
80	2480–2960	2640	63	672
60	2220–2530	2310	33	305
40	1960–2365	2050	24	162

V. CONCLUSION

The article introduced the upcoming IoT infrastructure for smart cities and putted it in the context of municipal waste management. We provided the summary on municipal waste collection management methods and showed the examples of solutions introduced by recent research in this area. Given overview showed that it is not yet enough discussed the possibility of using genetic algorithms as a optimization method for waste collection. Our solution is based on the idea of IoT infrastructure, which should provide enough information to handle this Smart City issue more efficiently.

The main part of the article provided experimental measurements and results, which proved the power of GA and also the possible cost savings (in average 15 %). The best parameters for the GA settings were shown (40 % throwing, 20 % max. crossover, population based on power requirements), when the low-power and power-independent real implementations were also considered.

ACKNOWLEDGMENT

The research described in this paper was financed by the National Sustainability Program under grant LO1401. For the research, infrastructure of the SIX centre was used.

REFERENCES

[1] P. Masek, K. Zeman, Z. Kuder, J. Hosek, S. Andreev, R. Fujdiak, F. Kropfl, *Wireless M-BUS: An Attractive M2M Technology for 5G- Grade Home Automation*, In EAI International Conference on CYber physiCaL systems, iOt and sensors Networks (CYCLONE 2015). Rome, 2015.

[2] ReTHINK, *Low power networks hold the key to IoT*, Rethink Technology Research Ltd. 2015.

[3] J. Hosek, P. Masek, D. Kovac, M. Ries, F. Kropfl, Universal Smart Energy Communication Platform. In 2014 International Conference on Intelligent Green Building and Smart Grid (IGBSG). 1. Taipei, Taiwan: IEEE, 2014. p. 1-4. ISBN: 9781467361217.

[4] M.N.K. Boulos and N.M. Al-Shorbaji, *On the Internet of Things, smart cities and the WHO Healthy Cities*, International Journal of Health Geographics. 2014.

[5] SIGFOX, *Keys for scalable M2M/IoT networks*, Tech. Slides presentation. 2014.

[6] A. Gallardo, M. Carlos, M. Peris and F.J. Colomer, *Methodology to design a municipal solid waste pre-collection system. A case study*, Waste Management (0956-053X), Vol. 36, p. 1–11. 2015.

[7] M. Minoglou and D. Komilis, *Optimizing the treatment and disposal of municipal solid wastes using mathematical programming - A case study in a Greek region*, Resources, Conservation and Recycling, Vol. 80, p. 46–57. 2013.

[8] E. C. Rada, *Energy from municipal solid waste*, In: 1st International Conference on Energy Production and Management in the 21st Century: The Quest for Sustainable Energy. Ekateringburg, Russian Federation, 2014.

[9] K. Mahajan and J.S. Chitode, *Waste Bin Monitoring System Using Integrated Technologies*, International Journal of Innovative Research in Science, Engineering and Technology (ISSN: 2319-8753), Vol. 3, Issue 7, p. 14953–14957. 2014.

[10] Y. Glouche and P. Couderc, *A Smart Waste Management with Self-Describing objects*, In: 2nd International Conference on Smart Systems, Devices and Technologies (SMART13). Rome, Italy, 2013.

[11] FarSite, *Bin Management System: Optimised bin monitoring and collection systems for smarter cities*, Datasheet v1.21. Ekateringburg, Russian Federation, 2014.

[12] N. Sharma, N. Singha and T. Dutta, *Smart Bin Implementation for Smart Cities*, International Journal of Scientific & Engineering Research, Vol.6 Issue 9. 2015.

[13] Link-Labs, *A Comprehensive look at Low Power, Wide Area Network*, For Internet of Things Engineers and Decision Makers. 2016.

[14] C. Chatzouridis and D. Komilis, *A methodology to optimally site and design municipal solid waste transfer stations using binary programming*, Resources, Conservation and Recycling, Vol. 60, p. 89–98. 2012.

[15] R. Minciardi, M. Paolucci and M. Robba, *Multi-objective optimization of solid waste flows: Environmentally sustainable strategies for municipalities*, Waste Management (0956-053X), Vol. 28, Issue 11, p. 2202–2212. 2008.

[16] M.F. Badran and S.M. Haggar, *Optimization of municipal solid waste management in Port Said Egypt*, Waste Management (0956-053X), Vol. 26, Issue 5, p. 534–545. 2005.

[17] Y. Li, F. James and R. Mei, *Performance characteristics of the diffusion driven desalination process*, Desalination, Vol. 196, Issue 1–3, p. 188–209. 2006.

[18] M.A. Najm, M. El-Fadel, G. Ayoub, M. El-Taha and F. Al-Awar, *An optimisation model for regional integrated solid waste management I. Model formulation*, Waste Management & Research, Vol. 20, Issue 1, p. 7–45. 2002.

[19] S. Rathi, *Optimization model for integrated municipal solid waste management in Mumbai, India*, Environment and Development Economics, Vol. 12, p. 105–121. 2007.

[20] E.O. Simonetto and D. Borenstein, *A decision support system for the operational planning of solid waste collection*, Waste Management (0956-053X), Vol. 27, Issue 10, p. 1286–1297. 2006.

[21] X.Y. Wu, L. Liu and J.B. Rathi, *An interval nonlinear program for the planning of waste management systems with economies-of-scale effectsA case study for the region of Hamilton, Ontario, Canada*, European Journal of Operational Research, Vol. 171, Issue 2, p. 349–372. 2006.

[22] S. Das and B. K. Bhattacharyya, *Optimization of municipal solid waste collection and transportation routes*, Waste Management (0956-053X), Vol. 43, p. 9–18. 2015.

[23] W. Xue, K. Cao and W. Li, *Municipal solid waste collection optimization in Singapore*, Applied Geography, Vol. 62, p. 182–190. 2015.

[24] S. Rathi, *Optimization model for integrated municipal solid waste management in Mumbai, India*, Environment and Development Economics, Vol. 12, p. 105–121. 2007.

[25] M. Chipumuro, R. Mawonike and T. Makoni, *Optimizing routing of residential solid waste collection: Case study of Chikova residential area in Zimbabwe*, International Research Journal of Mathematics, Engineering & IT (ISSN: 2349-0322), Vol. 1, Issue 3, p. 23–40. 2014.

[26] N.V. Karadimas, G. Kouzas, I. Anagnostopoulos and V. Loumos, *Urban Solid Waste Collection and Routing: The Ant Colony Strategic Approach*, International Journal of Simulation, Vol. 6, Issue 12–13, p. 45–53. 2005.

[27] Z.H. Ahmed, *Genetic algorithm for the traveling salesman problem using sequential constructive crossover operator*, International Journal of Biometrics & Bioinformatics (IJBB), Vol. 3 , Issue 6–13, p. 96–105. 2010.

[28] R. Fujdiak, J. Misurec, P. Mlynek and T. Petrak, *Verification of genetic algorithm in dynamic traffic light management*, In 38th Internaional Conference on Telecommunications and Signal Processing (TSP). 2015.

[29] S. Liu, *A Powerful Genetic Algorithm for Traveling Salesman Problem*, arXiv (1402-4699). 2014.

[30] Y. Deng, Y. Liu and D. Zhou, *An Improved Genetic Algorithm with Initial Population Strategy for Symmetric TSP*, Mathematical Problems in Engineering. 2015.

[31] F. Greco, *Travelling Salesman Problem*, I-Tech. Croatia, 2008.