Many-Objective Land Use Planning Using a Hypercube-Based NSGA-III Algorithm

Abstract: Due to the many objectives and constraints involved in urban land-use planning (ULUP), this is considered as a many-objective and complex optimization problem that needs a variety of geographical analyses. In this paper, the main target is improving NSGA-III as an advanced many-objective optimization algorithm for solving the ULUP problem. In this study, five objective functions (i.e., consistency, dependency, compactness, suitability, and per-capita violation of land-uses) are considered for simultaneous optimization for allocation. The proposed algorithm is tested using the spatial data of region 7, district 1 of Tehran using vector format. To evaluate the results, two more real datasets were implemented. The performance of the improved algorithm is compared concerning NSGA-II and NSGA-III in the main case study area and two other instances. The comparison results show that the improved algorithm increases the convergence and diversity of the generated solutions in ULUP concerning the results obtained by these two other algorithms. The results of optimization with these methods can help decision-makers towards sustainable development in the construction of new cities, new towns, and smart cities.

Keywords: Hypercube-based NSGA-III; many-objective optimization; Elite mutation for urban planning, Geospatial Information System (GIS), Sustainable development of urban areas

1. Introduction

The decision-making process in urban land-use allocation, as the main core of urban land-use planning (ULUP), allocates different urban land-uses to the suitable land units and results in the effective spatial organization of urban activities based on the demands and needs of urban society [1]. Generally speaking, the land-use planning process can be divided into two main associated parts [2]. The first part generates different scenarios based on defined objectives and constraints, while the second part aggregates stakeholders' preferences and selects a final land-use layout. The main core of land-use planning is the first part, which is concerned with allocating different land-uses to land units in order to achieve near-optimum layouts according to different objectives and constraints [2]. Indeed, an example of this multi-objective optimization problem is the allocation of different land-uses to various urban land units in order to produce a variety of alternatives, which satisfy multiple objectives and constraints. To solve this problem, classic optimization methods, such as linear programming (LP), are used in some research works [3, 4]. Heuristic and meta-heuristic methods are thus required to solve these NP-hard non-linear multi-objective optimization problems [5]. Two general approaches are considered to achieve optimal land-use layouts when tackling a multi-objective optimization problem, using heuristic or meta-heuristic methods. The first type of approach is based on determining the weights of the objectives prior to solving the problem and converting the multi-objective optimization problem into a single-objective optimization problem [2, 6, 7]. The second type of approach relates to the use of the Pareto front (PF) concept [8, 9].

For instance, in [10], goal programming, as one of the weighting methods, was applied to model the problem, while the genetic algorithm (GA) was used to settle it. To solve this optimization problem with the weighting approach, other heuristic or metaheuristic algorithms can be adopted, such as simulated annealing (SA) [11], ant colony optimization (ACO) [12], particle swarm optimization (PSO) [13], bee colony optimization (BCO) [14], hybrid methods [15], and so on. The main problems with weighting approaches are their dependency on the selected weights of the objective functions [16] and their inability to obtain non-convex portions of the Pareto front [17].

Besides, generating well-distributed PF for urban planners in selecting different land-use layouts with different objective trade-offs is impossible in these methods.

By considering the relative importance of each objective independently, the PF-based approach can overcome the aforementioned problems associated with the weighting approach. Accordingly, several studies have focused on solving a Multi-Objective Land-Use Optimization Problem (MOLUOP) based on the concept of PF [9, 17-19]. One of the early works based on this approach, Feng and Lin [19] used the Cumulative Genetic Algorithm (CGA) to generate alternative urban land-use layouts for urban planners. They proposed two objective functions to maximize the development efficiency and environmental harmony of land units. Moreover, Cao, Batty [17] generated various land-use allocation scenarios using the Non-Dominated Sorting Genetic Algorithm-II (NSGA-II). Their model consisted of three objectives, i.e., minimizing the land-use conversion cost, maximizing accessibility, and maximizing compatibility with land-uses in the neighborhood. Masoumi, Coello [20] employed NSGA-II for urban land-use planning. They considered the consistency, dependency, suitability, and compactness of land-uses as objective functions and also the per-capita demand as a criterion. They employed the original NSGA-II and also the GA operators (mutation and crossover).

Table 1 summarizes the limitations of different methods employed in the ULUP researches mentioned above. Moreover, the most common methods and their drawbacks are classified in Table 1. As can be seen in this table, the methods have been improving gradually to solve the ULUP problem in the real world. Recent researches employed multi-objective methods which are limited when the number of objectives increases. The NSGA-II algorithm proposed by Deb, Pratap [21] is one of the most widely-used approaches in spatial optimization and ULUP [17, 22, 23]. In NSGA-II, the crowding distance measure is defined and applied to maintain the diversity of solutions in objective space. NSGA-II was proposed to reduce computational complexity, improve the diversity of the solutions, and adapt appropriately with discrete search spaces [24]. Moreover, its relative ease of implementation [20, 25] is well known. Usually, MOLUOP depends on several objectives that should be considered during the optimization process. In this case (i.e. considering many objectives), the crowding distance tends to prefer dominance-resistant solutions, and it is not a suitable mechanism for selecting appropriate solutions to reach the true PF [26, 27]. To address the limitations of NSGA-II as a multi-objective optimization method, Deb and Jain [28] proposed NSGA-III, which is a many-objective optimization method and also get benefitted from the advantages of NSGA-II. In NSGA-III, the pre-defined reference points are responsible for preserving diversity. In NSGA-III, A hyper-plane must be generated to calculate the position of the reference points in objective space using the Das and Dennis systematic approach [29]. To generate this hyper-plane, it is essential to obtain the extreme points (maximum or minimum) near each axis in the objective space. In some cases, one extreme point may be near two different axes. If this happens, the generated hyper-plane has smaller dimensions than the objective space, which leads to abnormal results [30, 31]. Moreover, Ishibuchi, Imada [32] showed that in many-objective optimization problems, NSGA-III does not always generate better solutions than NSGA-II. On the other hand, due to the nature of spatial data employed in MOLUOP, the chromosome of the GA is very large and consists of several thousand genes. As such, the usual crossover and mutation operators adopted in the NSGA-III algorithm, inherited from a traditional GA, cannot effectively discover optimal solutions.

Table 1. The summary of literature review, methods they have used , and the drawbacks of methods

Main Methods	Sub-class methods	Authors	year	Drawbacks
Mathematical methods	Linear programming (LP)	Moah &Kanaroglou	2009	The method can not tackle non-linear objectives and constraints involved in complex problems
Converting Multi-objective to Single objective	Genetic algorithm (GA)	Porta et al.	2013	• The solutions depend on the selected weights of the objective functions • Inability to obtain non-convex portions of the Pareto front • Can not help the urban planners in selecting different land-use layouts with different objective trade-offs and the solution is unique
	Simulated annealing (SA)	Aerts & Heuvelink	2002	
	Ant colony optimization (ACO)	Liu, X., et al.	2012	
	Particle swarm optimization (PSO)	Shifa, et al.	2011	
	Bee colony algorithm(BCA)	Yang, et al	2015	
Multi-objectives optimization algorithm	Cumulative Genetic Algorithm (CGA)	Feng and Lin	1999	When the number of objectives increases, the diversity of solutions will be affected.
	Non-dominated sorting genetic algorithm-II (NSGA-II)	Cao, Batty	2015	
	Non-dominated sorting genetic algorithm-II (NSGA-II)	Masoumi et al	2020	

To overcome these drawbacks, in this work, instead of using reference points, hypercubes are relied upon to preserve the diversity of the solutions in a many-objective ULUP problem. When this method is employed, the objective space is first divided into different hypercubes. Following this, the solutions of each hypercube constitute a cluster. The number of neighbors of each solution within a cluster is considered a criterion to qualify the solution for the next generation. Solutions with a lower number of neighbors are likely to be chosen for the next generation. In addition to this, a new spatial-based mutation operator for ULUP, known as an elite mutation for urban planning (EMUP), is proposed to improve the ability to discover new solutions in objective space. By choosing elite solutions for each objective, the operator changes the land-use of some parcels to improve the fitness of the solutions in the neighborhood of the selected parcels. Therefore, the main target of this study is not to invent a new state-of-the-art multi-objective evolutionary algorithm; instead, we seek to improve the results of the NSGA-III as a many-objective optimization algorithm used for solving the ULUP problem. In specialized and complex problems such as ULUP in which the objective functions, input data, and problem details have a particular format and do not have a standard form, it is necessary to make significant changes to adapt the algorithms to the problem. Previous researchers, such as [15, 24, 30, 36, 37] have shown that in some cases the use of these algorithms in the ULUP problem does not create a diverse and well-ordered solution space. Therefore, the goal here is to balance the density of the solutions in the PF and to achieve a wider variety of solutions based on the use of the NSGA-III. To illustrate the functionality of the algorithm in solving the MOLOUP, the land-uses of district 1 in region 7 of Tehran are optimized taking into account five objectives, namely maximizing compatibility, dependency, compactness, suitability of land-uses and minimizing per capita demand violation. These objectives were selected based on the principle of urban land use design parameters [33]. Finally, the results of the proposed algorithm in MOLOUP are compared

with the results of NSGA-II and NSGA-III and also in two more case study areas to evaluate the efficiency of the proposed algorithm in this problem.

2. Spatial urban land-use allocation model

Land-use allocation is a complex process and presenting a comprehensive solution is not practical. The optimal allocation of urban land-uses is defined as a process of allocating various land-uses to different urban land units according to multiple criteria and constraints [20]. In this process, the size and shape of urban land units, the extent of neighborhoods, criteria and constraints can be very influential in the results.

The data format for urban land units can take the form of a regular raster grid or an irregular vector polygon layer. Each of these formats has its own advantages and disadvantages. Using the raster format facilitates the definition of the neighborhood, chromosomes encoding, and the application of GA operators because the computations can be conducted in a regular grid [10, 15, 22, 34-37]. However, the raster format is dependent on the size of grid cells (resolution). To model the details of land units using the raster format, the cell size must be small enough, which increases the number of cells and, consequently, the processing time as well. Using large cells also results in a loss of details, especially in the boundaries between urban land units. Consequently, the raster format does not model the actual shape of urban land units properly. On the contrary, in vector format, the urban land units are considered as polygons in their actual shape which corresponds to the reality of urban land units. Nevertheless, the vector format has some disadvantages, such as complexity in the neighborhood definition, failure in optimizing the parcels' shape, and maintaining the exact areas for various land-use types. However, using vector format results in precise boundaries between parcels. Furthermore, the vector format leads to lower computational load, which results in a reduction in the number of land units (as against grids) required to model urban parcels. As such, in this research, the vector format is used to model urban parcels. Figure 1 illustrates this subject in 3 different resolutions as 1, 5, and 10 meters. As seen in Figure 1, increasing the grid size resulted in missing the real shape of urban parcels in raster format but the shape of vector format is independent from the grid size.

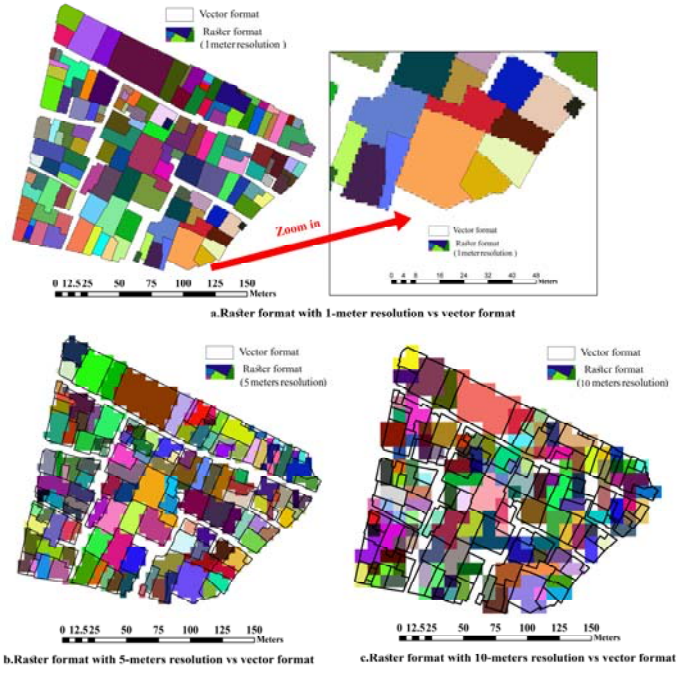


Figure 1. Raster format vs. Vector format in modeling land units in urban areas; a, b, and c parts show 1, 5, and 10-meter grid size in rater format in comparison with vector format respectively in a part of case study area

Several objectives and constraints can be considered when looking at the urban land-use allocation problem. The main objectives, which have been used in recent studies as neighborhood effects include compatibility, dependency, and compactness [9, 38-40].

The compatibility indicates no disturbance in adjacent land-uses. For example, residential and industrial land-uses are not compatible due to the noise and air pollution that industrial land-use produces. [9]. Dependency is used to express the dependencies between two land-use types. For instance, residential land-use requires commercial use nearby to meet everyday needs [33]. Equations (1) and (2) present the compatibility and dependency objective functions used in this study.

$$Compability = \left\{\frac{1}{P}\left(\sum_{i=1}^{P}\frac{1}{n_i}\sum_{j=1}^{n_i}\alpha_{ij}\times C_{ij}\right) + Min(\sum_{j=1}^{n_i}\alpha_{ij}\times C_{ij})\right\} \quad (1)$$

$$Dependency = \left\{\frac{1}{P}\left(\sum_{i=1}^{P}\frac{1}{n_i}\sum_{j=1}^{n_i}\alpha_{ij}\times D_{ij}\right) + Min(\sum_{j=1}^{n_i}\alpha_{ij}\times D_{ij})\right\} \quad (2)$$

where, i indicates the subject parcel and j indicates its neighbors; P is the number of all parcels, and n_i is the number of neighboring parcels of the ith parcel; C_{ij} is the compatibility between two land-uses of parcels i and j; D_{ij} is also the dependency between two land-uses of parcels i and j. The values of C_{ij} and D_{ij} are provided by experts' judgment about the compatibility and dependency of land-uses types gathered by the Delphi method in dependency and compatibility matrices. Because there are 11 main land-use types here (which are shown in Figure 4) and each of them has two or three subclasses only a part of the dependency matrix (e.g., between just two residential and commercial land-use types) is presented in Table 2.

Table 2. A part of the extracted dependency matrix from the Delphi model between residential and commercial land-uses, HD, MD, LD, MI, HI, and N refer to high dependency, medium dependency, low dependency, medium independence, high independence, and neutral respectively

Land-use type		Commercial			Residential		
		District	Regional	Daily	High density	Medium density	Low density
Residential	Low density	MD	MD	HD	LD	MD	MD
	Medium density	MD	MD	HD	MD	MD	
	High density	MI	MD	HD	MD		
Commercial	Daily	HD	MI	HI			
	Regional	HD	HI				
	District	HI					

These qualitative values in Table 2, were extracted by surveying among urban planning experts using the Delphi method. Some questionaries were designed first to give the experts' opinions. In the dependency matrix, HD, MD, LD, MI, HI, and N refer to high dependency, medium dependency, low dependency, medium independence, high independence, and neutral respectively which is directly asked from experts in the questionaries. In the next stage, the conflicts between expert's opinions were extracted and summarized with them in a meeting. Three rounds of surveying are done in this research to solve some conflicts in expert's views. This is a common process followed in land-use planning to extract these matrices.

Then, since the algorithm works with numerical values, the qualitative values obtained from the Delphi are converted to numerical values using the structured pairwise comparison based on Analytic Hierarchy Process (AHP). In this method, the process of comparing the decision criteria is done in two stages. First, each criterion is ranked based on its importance and priority, and then the two adjacent criteria are compared. For example, since the proximity of two dependent land-uses increases their usefulness and the proximity of two independent users reduce their usefulness, negative effects and consequences should be more important. So, the change from "N" level to "MD" level is considered more important than a change from "N" level to "MI". Table 3 shows the degree of importance of the different levels of dependency in comparison with each other.

Table 3. Structured pairwise comparison between decision criteria

Dependency levels based on the Delphi method	The relative importance of two levels of dependency
HD to MD	More important than= 1 level
MD to N	More important than= 1 level
N to MI	Much more important than= 2 levels
MI to HI	Much more important than= 2 levels

Table 4 also shows how to determine the relative importance of each degree of dependence pair and calculate the numerical values for them using AHP.

Table 4. The values obtained for levels of dependency using the AHP method

Dependency levels	HD	MD	N	MI	HI	Geometric mean	Standardized value
HD	1	2	3	5	7	2.9137	0.43
MD	0.5	1	2	4	6	1.8882	0.28
N	0.33	0.5	1	3	5	1.2011	0.18
MI	0.2	0.25	0.33	1	3	0.5492	0.08
HI	0.14	0.17	0.2	0.33	1	0.2756	0.04

The compatibility matrix was produced in the same structure but here the matrix was filled with the consistency values. For more information about construction details of these matrices and the AHP method, the readers can refer to [41].

In equations (1) and (2), α_{ij} represents a distance decay function that controls the impact of distance in the value of compatibility and dependency between two parcels i and j which is defined as Equation (3) [42].

$$\alpha_{ij} = \begin{cases} \left(\frac{d^k_{max} - d_{ij}}{d^k_{max}}\right) & d_{ij} \le d^k_{max} \\ 0 & d_{ij} > d^k_{max} \end{cases} \qquad (3)$$

where d^k_{max} is the maximum distance that land-use k can affect on other land-uses, and d_{ij} is the nearest Euclidean distance between two parcels i and j. The value of d^k_{max} is obtained from the radius of effect table that is provided by. This function is used to reduce the effect of land-use types on each other by increasing the distance between two land units. The larger the distance, the less the effect of two parcels is.

The second part of Equations (1) and (2) is used to maximize the minimum value of the objectives in the solutions.

The compactness objective is used to measure the similarity of neighboring land-uses in a neighborhood. In urban land-use planning, most land-uses tend to be located in areas that have similar land-uses. Hence, the compactness objective function is defined in Equation (4)[43].

$$Compactness = \frac{1}{P}\left(\sum_{i=1}^{P}\frac{sim_i}{n_i}\right) \quad (4)$$

In Equation (4), sim_i is the number of neighboring parcels with a land-use type similar to the ith parcel. Physical suitability is also an important objective in the land-use allocation process. This criterion indicates the degree of physical and environmental adaptation of a land-use type to different land units [14]. Physical suitability is dependent on different parameters, such as the size and dimensions of the land unit, access to the city's public transport network, air pollution, noise pollution, and the difficulty in changing to another land-use type [9]. By using a weighted combination of these parameters, we can estimate the suitability of each land-use for parcels of the study area. The overall suitability of the entire study area is calculated using Equation (5).

$$suitability = \left\{\frac{1}{P}\left(\sum_{i=1}^{P}S_i\right) + Min(S_i)\right\} \quad (5)$$

where, S_i is the suitability of parcel i concerning its current land-use. The second part of this equation is also used to maximize the lowest value of suitability in the area.

The requirement of each person for a certain area of any land-use type is defined as per-capita demand in urban land-use planning which is defined as a standard value in a detailed city plan. Violations of standard per capita demand occur in two situations: more than per capita and less than per capita. In this research, per capita violation is assumed as a penalty function to suppress invalid solutions [44]. The optimization process focuses on minimizing the amount of per capita demand violation (PCV). Therefore, in this study, PCV is considered an objective function to be minimized in the optimization process. The PCV is calculated using Equation (6).

$$PCV = \sum_{k=1}^{U} V_k \quad (6)$$

where U indicates the number of land-use types, and V_k is the per capita demand violation of the land-use k, which is calculated using Equation (7).

$$V_k = \begin{cases} \dfrac{minA_k - A_k}{minA_k} & if\ A_k < minA_k. \\ \dfrac{A_k - maxA_k}{maxA_k} & if\ A_k > maxA_k. \\ 0 & if\ minA_k \le A_k \le maxA_k. \end{cases} \quad (7)$$

where $minA_k$ and $maxA_k$ indicate the minimum and the maximum acceptable area of the land-use type k and A_k is the current area of the land-use type k. The $minA_k$ and $maxA_k$ are obtained from [45]. Table 5 briefly describes the objective functions and their definition.

Using regulatory knowledge in the form of constraints can avoid the generation of non-real solutions in urban land use planning [2]. A sample constraint derived from regulatory knowledge is that "land-units must have access to the transportation network appropriate to their land-use type". These constraints are applied to allocate valid land-uses to parcels in two steps: the initial solution generation step, and the mutation step, both of which are discussed in Section 3.1 and Section 3.2, respectively.

Table 5. An overview of the objective functions and their brief definition in the problem in land-use arrangements

Objective functions	Formula	Main component of formula	Brief description	Target (Minimizing/Maximizing)
Compatibility	$Compability = \left\{\frac{1}{P}\left(\sum_{i=1}^{P}\frac{1}{n_i}\sum_{j=1}^{n_i}\alpha_{ij}\times C_{ij}\right) + Min(\sum_{j=1}^{n_i}\alpha_{ij}\times C_{ij})\right\}$	C_{ij}: compute compatibility using its derived matrix $\alpha_{ij:}$ model distance effect on compatibility	Measures consistency between neighbor land-use types	Maximization
Dependency	$Dependency = \left\{\frac{1}{P}\left(\sum_{i=1}^{P}\frac{1}{n_i}\sum_{j=1}^{n_i}\alpha_{ij}\times D_{ij}\right) + Min(\sum_{j=1}^{n_i}\alpha_{ij}\times D_{ij})\right\}$	C_{ij}: compute dependency using its derived matrix $\alpha_{ij:}$ model distance effect on dependency	Measures requirements of land-uses to each other	Maximization
Suitability	$suitability = \left\{\frac{1}{P}\left(\sum_{i=1}^{P}S_i\right) + Min(S_i)\right\}$	S_i: suitability of each land unit for land-use types	Measures the physical suitability of a land with its land-use type	Maximization
Compactness	$Compactness = \frac{1}{P}\left(\sum_{i=1}^{P}\frac{sim_i}{n_i}\right)$	Sim_i: a counter to compute the similar land-use types in neighbors	Avoids scattering in land use types of neighbor parcels	Maximization
Per capita violation	$PCV = \sum_{k=1}^{U} V_k$	V_k: compute the difference between per capita demand in an arrangement and standard per capita demand	Prevents land-use arrangements with high or low per capita demand in comparison with standard values	Minimization

The application of this type of problem in the real world is firstly when urban land use planners want to see the impact of land-use change in an area of the city. In an urban system, the demand for a change of land-uses must be constantly examined by experts. External consequences mean how much the balance between urban uses is disturbed in terms of the five objective functions provided in Table 5. The calculation of this case was previously done by experts at the neighborhood level, but the real effect is beyond the neighborhood [46]. In recent years, this effect has been calculated approximately using multi-criteria decision systems [34, 40]. This concept is especially important in the smart cities' context, which is one of the most advanced urban concepts nowadays. Recent research in this field seeks to use optimization algorithms and adapt them to solve the ULUP problem. That is, if land-use changes, what will be the optimal arrangement of other land-users by considering all objective functions simultaneously? Therefore, once the optimal layout is determined, the experts will be able to observe the changed land-uses and decide whether the costs of this change are acceptable or not for the region and the whole city.

Furthermore, the optimized arrangement of land-uses is desirable in cases such as designing new cities, city development plans, and sketching the detailed plan of a city. In these cases, the urban planner should arrange the different land-uses in the urban areas considering the above-mentioned objectives simultaneously [46]. Planners, therefore, face the challenge of having to deal with many objectives, and multi-objective optimization offers an option to balance potential trade-offs among the objectives allowing to achieve sustainable development in the cities [39]. Overall, this and other recent studies in this field are raising the interest in adopting state-of-the-art multi- and many-objective optimization algorithms for solving the ULUP problem.

As discussed before, in the first application, it is necessary to know the legal boundaries between the parcels. So, it is essential to use vector data that are well able to display these boundaries. However, in the second case, raster data can be used. Still, the results would be much more accurate if the initial boundaries of the zones or parcels were taken into account (which is possible using vector data).

3. Hypercube-Based NSGA-III for MOLUOP

In this section, a many-objective optimization algorithm based on NSGA-III is proposed to solve MOLUOP.

In the proposed algorithm, instead of using reference points, a grid is employed to preserve the diversity of solutions. In the many-dimensional objective space, this grid becomes a set of hypercubes to cluster and maintain the diversity of solutions. Due to the use of hypercubes to preserve diversity, the proposed algorithm is called Hypercube-based NSGA-III (HNSGA-III).

3.1. HNSGA-III

The proposed algorithm is similar to NSGA-III, having its main differences related to diversity maintenance, next-generation selection, and its mutation operator (EMUP).

The proposed algorithm has an iterative process to optimize the initial population by preserving diversity of the solutions. At each iteration, in the first step, the offspring are generated from parents by applying crossover, mutation and our proposed EMUP operator. In the next step, the total population is evaluated and sorted into different non-domination levels. The non-dominated sorting algorithm acts similarly to the one used in the NSGA-III algorithm, and is carried out based on dominance rules [47]. In the last step, N (number of generations in each iteration) solutions must be selected from the total population. In order to preserve Elite-members, the algorithm starts to select solutions from the first non-domination level and continues until the number of the next generation (G_{t+1}) becomes greater than or equal to N. If G_{t+1} becomes exactly equal to N, there is no need to perform other operations, and the algorithm goes to the next iteration. Otherwise, the surplus solutions must be removed from the last front that is selected to contribute in the next generation. For this purpose, the number of neighbors of each solution is calculated by constructing hypercubes in objective space and assigning each solution to a hypercube. On the last front, the chance of removing

solutions with more neighbors is higher. The procedure of producing a generation is illustrated in Algorithm 1.

Algorithm 1: The procedure of HNSGA-III
Input: The population of the generation t: P_t
Output: The population of generation $t+1$: P_{t+1}
$Ch_{t+1} = Crossover + Mutation(P_t)$
$T_{t+1} = P_t + Ch_{t+1}$
$Fr[\,] = Non\_dominated\_sort(T_{t+1})$
$G_{t+1} = [\,].\, Fr_L = [\,]$
$i = 1$
repeat
  $G_{t+1} = G_{t+1} \cup Fr[\mathrm{i}]$
  $i = i + 1$
until $|G_{t+1}| \geq N$
if $(|G_{t+1}| = N)$ **then** $P_{t+1} = G_{t+1}$. **break**;
else
  $P_{t+1} = \cup_{j=1}^{i-1} Fr[j]$
  $Fr_L = Fr[i]$ % Fr_L: The last selected front for the next generation.
  $CalNoN(\,Fr_L)$ % $CalNoN$ is detailed in Algorithm 2.
  $K = N - |P_{t+1}|$ % K: The number of solutions that remain in the Fr_L.
  $R_L = RDH\,(Fr_L.K)$ % RDH is detailed in Algorithm 4.
  $P_{t+1} = P_{t+1} \cup R_L$
end if

In Algorithm 1, Ch_{t+1} is the generated offspring for generation $t+1$; T_{t+1} indicates the combined parent and offspring population; $Fr[\,]$ maintains the fronts; $Fr[\mathrm{i}]$ is the i^{th} front; G_{t+1} is the candidate solutions for the next generation; Fr_L is the last front that is selected to contribute in the next generation and R_L contains the remaining solutions from Fr_L to be included in the next generation (P_{t+1}).

3.1.1. Calculating the number of neighbors of the solutions

To count the number of neighbors of each solution, the hypercube of each solution must be determined based on its position in objective space. After determination of the hypercube of each solution, the hypercube ID (HID) is assigned to the HID property of each solution. HID is the identification number of the hypercube. All of the solutions that have the same HID are the neighbors of each other. Algorithm 2 shows the procedure of calculating the number of neighbors of each solution.

Algorithm 2 ($CalNoN$): Calculating the number of neighbors of the solutions
Input: Last front without the number of neighbors of each solution: Fr_L
Output: Last front with the number of neighbors of each solution: Fr_L
1: $FindHID(Fr_L)$ % Detailed in Algorithm 3
2: **for** $i = 1$ **to** $|Fr_L|$
3: **for** $j = i + 1$ to $|Fr_L|$
4: **if** ($Fr_L[i].HID == Fr_L[j].HID$)
5: $Fr_L[i].NoN$++;
6: $Fr_L[j].NoN$++;
7: **end if**
8: **end for**
9: **end for**

In Algorithm 2, $Fr_L[i].HID$ is the hypercube *ID* of the i^{th} solution in Fr_L; $Fr_L[i].NoN$ indicates the number of neighbors of the i^{th} solution in Fr_L.

The procedure of finding and assigning HID to each solution is shown in Algorithm 3. The hypercubes of Fr_L is generated in lines 1 to 5 of Algorithm 3. In these lines, first, the minimum and maximum values of each objective are determined. Then, according to the minimum and maximum values and number of divisions (D), the width of the hypercubes is calculated along each objective axis. In lines 6 to 14 of Algorithm 3, the *HID* of all solutions in Fr_L are determined considering their position in objective space. Figure 2 illustrates a schematic view of hypercubes and their HID in three dimensions.

Algorithm 3 (*FindHID*): Finding and assigning HID to each solution in Fr_L.
Inputs: Last front: Fr_L
Outputs: Last front with HID property
1: **for** $i = 1$ **to** M
2: $Max[i]$ = Find maximum value of objective i in Fr_L;
3: $Min[i]$ = Find minimum value of objective i in Fr_L;
4: $W[i] = (Max[i] - Min[i])/D$
5: **end for**
6: **for** $j = 1$ **to** $|Fr_L|$
7: **for** $i = 1$ **to** M
8: **for** $k = 1$ **to** D
9: **if** $(Min[i] + k * W[i]) \leq Fr_L[j].obj[i] < (Min[i] + (k+1) * W[i])$
10: $Fr_L[j].HID[i] = k$;
11: **end if**
12: **end for**
13: **end for**
14: **end for**

In Algorithm 3, M is the number of objectives; $Max[i]$ and $Min[i]$ are the maximum and minimum values of the i^{th} objective function; $W[i]$ is the width of the hypercubes along the i^{th} objective axis; D indicates the number of divisions in each objective; $Fr_L[j].obj[i]$ refers to the i^{th} objective value of the j^{th} solution in Fr_L, and $Fr_L[j].HID[i]$ refers to HID of the i^{th} objective of the j^{th} solution in Fr_L.

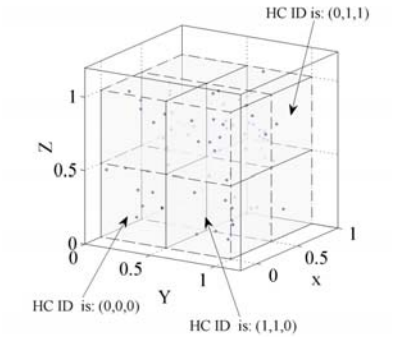


Figure 2. A schematic view of generated hypercubes in 3D and their *HID*.

3.1.2. Removing surplus solutions from the last front

An iterative tournament selection procedure picks and removes surplus solutions from Fr_L. This procedure is detailed in Algorithm 4. In this procedure, after picking a solution with a high number of neighbors, NoN of the neighbors of the picked solution are decreased by one unit and the selected solution will be removed. This procedure continues until all surplus solutions are removed from the last front.

Algorithm 4 (*RDH*)**:** Removing solutions from dense hypercubes
Inputs: last front: Fr_L
Number of solutions that must be remained in Fr_L: K
Output: The remained solutions in Fr_L: R_L
1: **for** $i = 1$ **to** $|Fr_L| - K$
2: select a solution to remove: $Sol = Tournoment\_Selection(Fr_L)$
3: **for each** (neighboring solution of Sol: Sol_n)
4: $Sol_n.NoN = Sol_n.NoN - 1$
5: **end for**
6: remove Sol from Fr_L
7: **end for**

In Algorithm 4, $Sol_n.NoN$ indicates the number of neighbors of the neighboring solution (Sol_n). It should be noted that, if the number of neighbors in the tournament selection is the same, one of the solutions will be deleted randomly, because they are not different and they are both at the same level of importance in the algorithm.

3.2. Genetic Operators in HNSGA-III

It is necessary to redesign the structure of chromosomes and genes as well as the GA operators of the algorithm to be compatible with MOLUOP. Hence, chromosomes represent land-use layouts and genes represent parcels. Each parcel has a value that indicates its land-use type. GA operators, including initialization, crossover, mutation, and EMUP (for MOLUOP) are explained in the following subsections.

3.2.1. Initialization

In most studies related to optimal land-use allocation, to achieve a valid solution, all or some of the solutions of the initial population are generated by applying a random modification to the current land-use layout in the study area. This can reduce the efficiency of searching for optimal solutions in the entire solution space. Consequently, to generate the initial population independent of the status quo, initial land-use layouts are randomly generated only concerning access type and area constraints. To assign a valid land-use to parcels, first, valid land-uses for each parcel are selected in a way that both constraints are satisfied. Then, from among the valid land-uses, one is randomly assigned to the parcel. The pseudo-code of generating an initial random solution is shown in Algorithm 5.

Algorithm 5: Generating an initial random solution considering the constraints.

Input: Parcels of the study area: $Parcels$

Output: An initial random solution: Sol

for $i = 1$ **to** $|Parcels|$
  VLU = **select** land-uses **from** LUs
    where $LUs.minArea < Parcels[i].Area$,
    and $LUs.maxArea > Parcels[i].Area$,
    and $LUs.ValidAccessTypes.Contains(Parcel[i].AccessType)$;
  $Parcel[i].Landuse$ = Select a random land-use from VLU;
  if (VLU is empty)
    $Parcel[i].Landuse$ = Select a random land-use from LUs;
  end if
  $Sol = Sol \cup Parcel[i]$;
end for

In Algorithm 5, LUs refers to the list of all land-uses; VLU represents the valid land-uses for parcel i; $Parcel[i].Landuse$ indicates the land-use of the i^{th} parcel in $Parcel$ set; $Parcels[i].Area$ indicates the area of parcel i; $LUs.minArea$ and $LUs.maxArea$ is the list of the minimum and maximum valid parcel areas of land-uses in LUs and $LUs.ValidAccessTypes$ is the list of valid access types of land-uses in LUs.

It should be noted that approximately 1.5% of parcels may have an invalid LU. Because the number of these parcels is so rare, we used a random assignment of an LU to them.

3.2.2. Crossover

The crossover operator generates two offspring by selecting two parents using tournament selection. This operator randomly divides both parents into N_p parts to generate offspring. N_p is a random number between 2 and 100. This means that the operator divides the parents into a minimum of 2 parts and a maximum of 100. The operator then generates two offspring by swapping these parts between parents. Algorithm 6 shows the procedure of the crossover operator. Moreover, Figures 3(a) and (b) represent the structure of a chromosome and an example of the crossover operator for N_p=5 respectively in this research.

Algorithm 6: Crossover operator
Inputs: Two parents: P_1 and P_2.
Outputs: Two offspring: O_1 *and* O_2.
1: N_p = Select a random number between 2 and 100;
2: **for** $i = 1$ **to** N_p;
3: **if** i *is Odd*
4: $O_1[i] = P_2[i]$;
5: $O_2[i] = P_1[i]$;
6: **else**
7: $O_1[i] = P_1[i]$;
8: $O_2[i] = P_2[i]$;
9: **end if**
10: **end for**

389

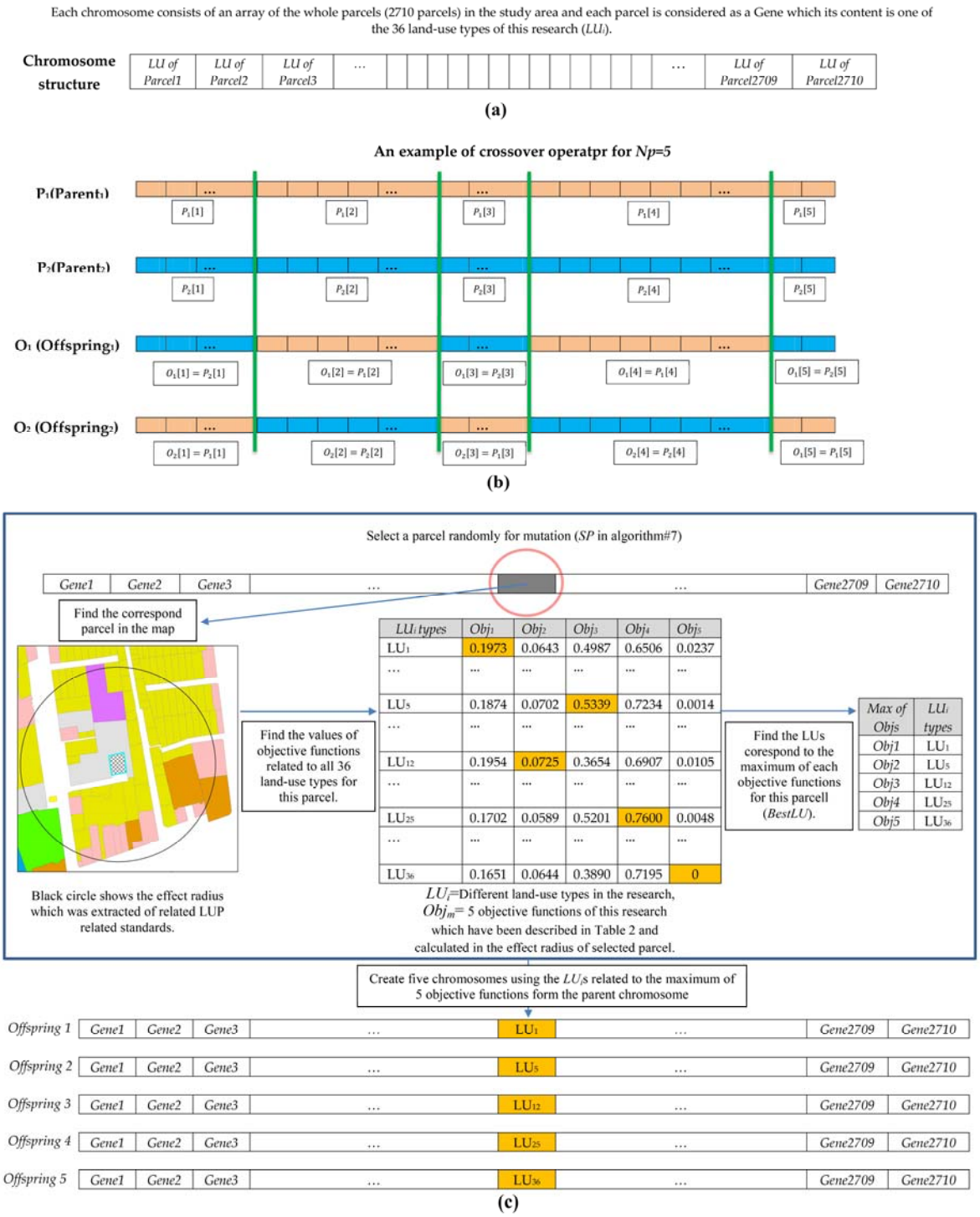
390 3.2.3. Mutation

391 In addition to the conventional mutation in which 10% of the parcels are randomly selected and
392 their land-uses are randomly changed to another valid land-use, a new mutation operator which we
393 call here EMUP, is proposed as part of our approach. This operator attempts to improve the objective
394 functions locally in elite solutions by selecting randomly 10% of the parcels of these solutions and
395 assigning the best local land-use for these parcels. At first, the algorithm finds the neighbors of the
396 selected parcels in the map. Then, the objective functions for all possible land-use types for these
397 parcels are calculated. The best land-use for objective i on a selected parcel is determined by
398 calculating and sorting the fitness of objective i for all the valid land-uses. The constraints are also
399 applied to determine valid solutions for a parcel. The details of the EMUP operator are illustrated in
400 Algorithm 7 and Figure 3 (c). In Figure 3 (c), the steps of EMUP are described schematically for just
401 one parcel. Actually, in the proposed algorithm this operator is applied to 10% of the parcels.

Algorithm 7: EMUP for an elite solution
Inputs: Elite solution: ES
Outputs: Elite Offspring: EO
1: **for** $i = 1$ **to** M
2: SP= Select randomly 10% of parcels from (ES);
3: **for each** parcel P **in** SP
4: $VLUs[P]$ = select valid land-uses for (P);
5: $BestLU[P]$ = current land-use of P;
6: **for each** LU **in** $VLUs[P]$
7: $Fitness$ = Calculate local fitness of LU in P;
8: **if** ($Fitness$ is better than the fitness of $BestLU[P]$)
9: $BestLU[P] = LU$;
10: **end if**
11: **end for**
12: **end for**
13: **end for**

402

403 In Algorithm 7, M is the number of objectives; SP is the list of the 10% of all parcels that are
404 selected randomly, and $BestLU[P]$ indicates the best land-use for parcel P.



LUi types	Obj1	Obj2	Obj3	Obj4	Obj5
LU1	0.1973	0.0643	0.4987	0.6506	0.0237
…	…	…	…	…	…
LU5	0.1874	0.0702	0.5339	0.7234	0.0014
…	…	…	…	…	…
LU12	0.1954	0.0725	0.3654	0.6907	0.0105
…	…	…	…	…	…
LU25	0.1702	0.0589	0.5201	0.7600	0.0048
…	…	…	…	…	…
LU36	0.1651	0.0644	0.3890	0.7195	0

Max of Objs	LUi types
Obj1	LU1
Obj2	LU5
Obj3	LU12
Obj4	LU25
Obj5	LU36

Figure 3. Definition of chromosomes, crossover, and mutation stages in the proposed algorithm, (a) shows the chromosome structure, (b) describes an example of the crossover operator, (c) indicates the stages of the mutation operator

4. Study Area and Implementation Requisites

The land-use map of region 7, district 1 of Tehran at the parcel level on the scale 1:2000 is used to implement and test the proposed algorithm. The presence of various land-uses at different service levels is the most significant matter when it comes to this study area, which complicates the issue of urban land-use planning. The parcels layer is applied in the form of vector data and 11 urban land-use types are defined with respect to the three service levels: local, district, and regional. In this research, an attempt has been made to provide completely real data from the study area and the

objective functions. For this purpose, the existing urban design standards in Iran and related documents have been used. The number of urban parcels in the study area is 2,710, which covers an area of approximately 52 ha. Figure 4 shows the urban parcels and their land-uses in the study area.

Figure 4. Land-uses of the study area

The residential density concerning the area of the residential units is extracted from the development pattern of Tehran [48]. The residential density is used in the calculation of the per capita demand for residential use. Urban land-use access types are also extracted from the development pattern of Tehran [48] and are used in applying the constraints related to access types. The parcel area ranges for each land-use are extracted from [49] and are used in applying the constraints related to parcel area. The implementation of the proposed algorithm was carried out in Visual Studio 2010 using C#, and was executed on a computer system with an Intel Core i5-4200U processor, and 4GB of RAM.

5. Experiments and results

In this section, the parameter selection procedure of the algorithm is first discussed. Following this, the effect of the EMUP operator is investigated. Finally, an analysis of the optimization results is presented. All the algorithms in this study are independently run 10 times on each case study problems and related results like HV indicator, DCI, and effect size values reported in this section.

5.1. Parameter selection

To determine optimal values for the parameters, the proposed algorithm is executed with different parameters 10 times, and the best value for each parameter is selected.

The HV is a popular performance indicator that is commonly used to assess the performance of different evolutionary approaches in multi- and many-objective optimization methods; it assesses the results in terms of convergence and maximum spread [50]. HV has also been employed for the selection of initial parameters in evolutionary multi- and many-objective optimization algorithms. This indicator measures the volume enclosed between a reference point (usually the worst objective function values) and a PF approximation. A greater value indicates that the solutions are more distant from the reference point and more scattered in objective function space (i.e., we have a better PF approximation). For more information on this indicator, see [51, 52]. The population sizes considered were 250, 500, 750, and 1000, and the maximum number of iterations of the algorithm was set as 500.

Figure 5 shows the changes in HV values for the initial population sizes of 250, 500, and 750 with a maximum number of iterations of 500. As can be seen in this figure, the HV index when using a population size of 250 had not converged after 500 iterations, and the HV value in this case is lower. In contrast, with population sizes of 500 and 750, this algorithm converges in about 150 iterations, and the values of the HV indicators for these two executions are almost the same. As a result, we concluded that the population size required to achieve relative stability is 500. For higher accuracy, the population size and the maximum number of iterations considered were 600 and 200, respectively. In addition, after considering the various executions, the crossover and mutation rates were set as 0.6 and 0.4, correspondingly. It should be noted that the run with the initial population size of 1000 and a maximum number of iterations of 500 had almost the same HV results as those obtained with the previously indicated settings, but the execution time in this case, was significantly higher.

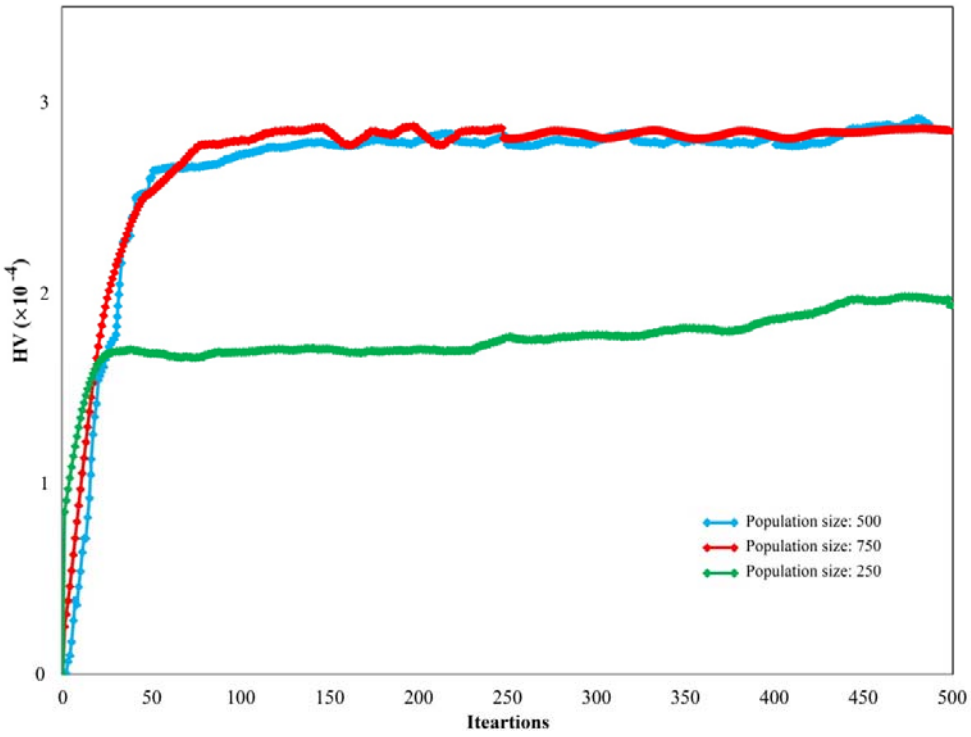


Figure 5. The HV indicator with respect to different population sizes.

The number of hypercubes is an important parameter in preserving the diversity of the solutions in the PF. To determine the suitable value of this parameter, the algorithm is executed by taking into account the previous parameters and assigning different values to the number of hypercubes used to assess the diversity metric (*DM*). *DM* is an indicator that examines the range searched by the algorithm. The more the algorithm searches for the larger range, the more varied are the optimal solutions, and, consequently, the better is the performance [53]. If $f_m^{\min}$ and $f_m^{\max}$ are indicative of maximum and minimum values for the m^{th} objective function, respectively, Δf_j are defined using Equation 8 [54].

$$\Delta f_j = f_j^{\max} - f_j^{\min}, j = 1,2,...,m \quad (8)$$

where m is the number of objective functions. The ΔF vector is also defined using Equation 9.

$$\Delta F = \left[\Delta f_1, \Delta f_2, ..., \Delta f_m\right] \quad (9)$$

The diversity metric is then calculated using Equation 10.

$$DM = \|\Delta F\| \quad (10)$$

where, the $\|\Delta F\|$ is the second norm of ΔF.

As shown in Figure 6, the results of the different executions show that the *DM* [55] of the results improves with an increase in the number of divisions on each dimension in the objective space. This improvement continues until the number of divisions reaches ten, although after ten divisions no tangible improvements are found in *DM*. The *DM* becomes worse as the number of divisions rises above 30. By increasing the number of divisions of each dimension in objective space, the number of hypercubes increases exponentially. Thus, most of the hypercubes are empty, and just some hypercubes contain one or two solutions. In this case, the neighbor count of each solution falls to zero or one, and therefore the importance of all solutions is nearly equal in terms of diversity. In other words, the algorithm selects the next generation randomly and the importance of each solution does not influence the selection process. In this situation, the results are facing a reduction in the *DM*. As a result, the suitable value for the number of divisions is ten in this research.

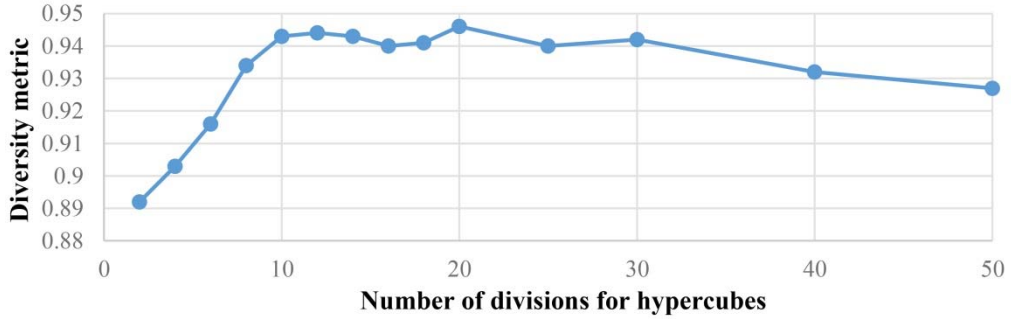


Figure 6. Diversity metric of results with respect to the different number of divisions in each dimension of objective space.

5.2. The effect of the EMUP operator on the results

One of the most important parts of the proposed algorithm is the EMUP operator. To evaluate the effectiveness of this operator, the algorithm is executed ten times, with and without EMUP, and the average of the worst and the best values of each set of PF together with the respective HV values for two study cases are calculated. Table 3 compares the worst, the best and the extent (the difference between the worst and the best values) from the ten executions. The results of the comparison show that the presence of this operator can significantly improve the results. In addition, the interval between the worst and the best values of the objective functions exhibits a considerable increase when this operator is applied, which represents a significant spread in the approximated PF.

Table 3. The average of worst/best, and the extent of objective values from the ten executions with and without EMUP.

	Without EMUP			**With EMUP**		
	Worst	Best	Extent	Worst	Best	Extent
Objective 1	0.1613	0.1657	0.0044	0.1036	0.1973	0.0937
Objective 2	0.0461	0.0473	0.0012	0.0396	0.0725	0.0329
Objective 3	0.4376	0.4619	0.0243	0.1426	0.5339	0.3913
Objective 4	0.4298	0.4721	0.0423	0.2511	0.7600	0.5089
Objective 5	0.7023	0	0.7023	9.725	0	9.725

Figure 7 presents the spread of the approximated PF with and without the EMUP operator along with the HV indicator of the two study cases. In Figure 7 (a), the approximated PF is shown in three dimensions (compatibility, dependency, and compactness) for both cases. The algorithm detects only a small part of the PF when the EMUP operator is not applied. In this case, all of the non-dominated solutions are placed in a part of objective space, which is magnified in Figure 7(a) for a better view. But by applying the EMUP, the spread of results, with both worst and best values significantly increases and the algorithm can find new solutions and act in a more exploratory way. Figure 7 (b)

also presents the HV indicator of the approximated PF with and without the EMUP operator. The comparison between two HV diagrams indicates that the HV of the PF when incorporating EMUP is extensively (1000 times) larger than the case without EMUP. Furthermore, the algorithm that incorporates EMUP converges in approximately 50 iterations. Conversely, the algorithm without EMUP converges in about 150 iterations. So, the algorithm with EMUP achieves convergence in fewer iterations.

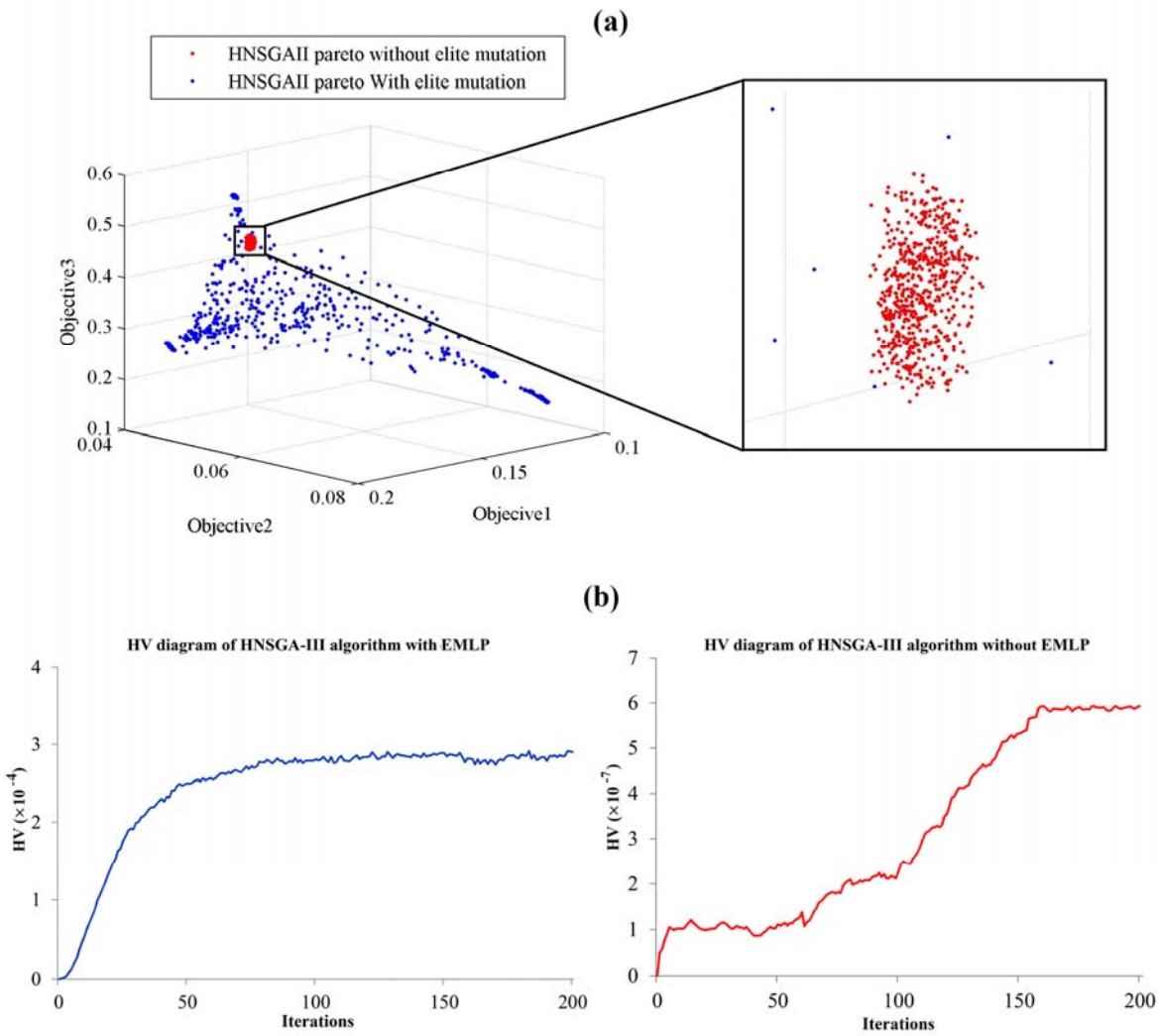


Figure 7. Approximated PFs along with the HV diagram in 200 iterations both when using the EMUP operator and without using it, (a). The related PFs in three dimensions, (b) the HV diagrams.

5.3. Analysis of the optimization results

Figure 8 shows the approximated PF in the first, tenth and last iterations. In this figure, the improvement of the solutions is illustrated during the optimization process. The figure also shows that the initial solutions are not distributed well in objective space and are focused in a small area. In the next iterations, the solutions are well-distributed and a wide range of the space is covered. In addition, the objective values are improved in the next iterations.

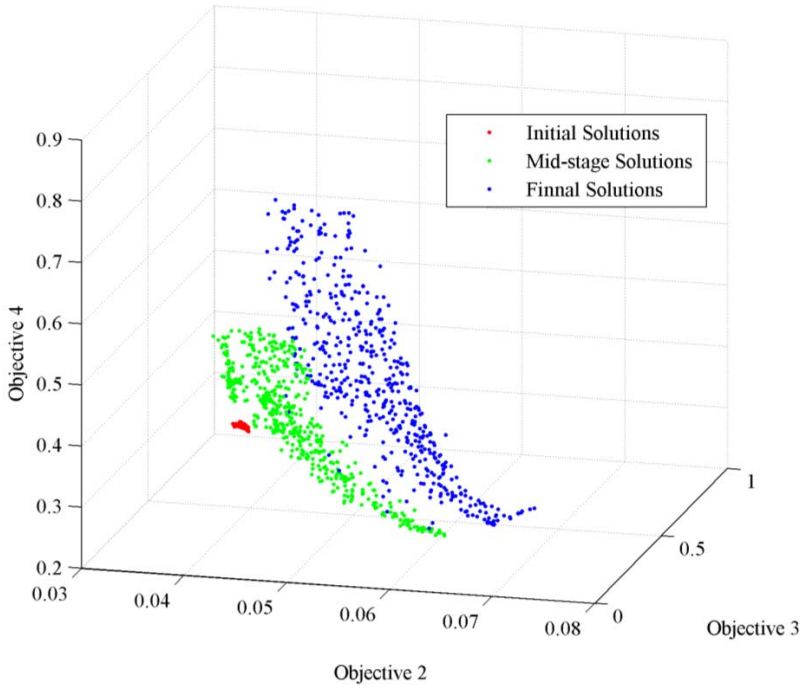


Figure 8. Approximated PF in different iterations

An analysis of the results from ten different executions of the algorithm shows that the repeatability of the algorithm is in a suitable condition. As shown by Figure 9, in ten different executions with the same parameters, the best solutions regarding the first objective, 93% of the parcels receive the same land-use. In addition, in terms of the best solutions regarding the second, third and fourth objectives, 89.5%, 99.5%, and 88.2% of parcels respectively receive the same land-uses in ten executions. However, the situation is different for the fifth objective, where only 13% of the parcels receive the same land-uses in the different executions. This is due to the nature of the fifth objective function. The fifth objective function is calculated as a violation of the minimum and maximum per capita demand, and for this reason various solutions can be without per capita violation; indeed, this variety in solutions leads to only 13% of parcels receiving the same land-uses in all ten runs.

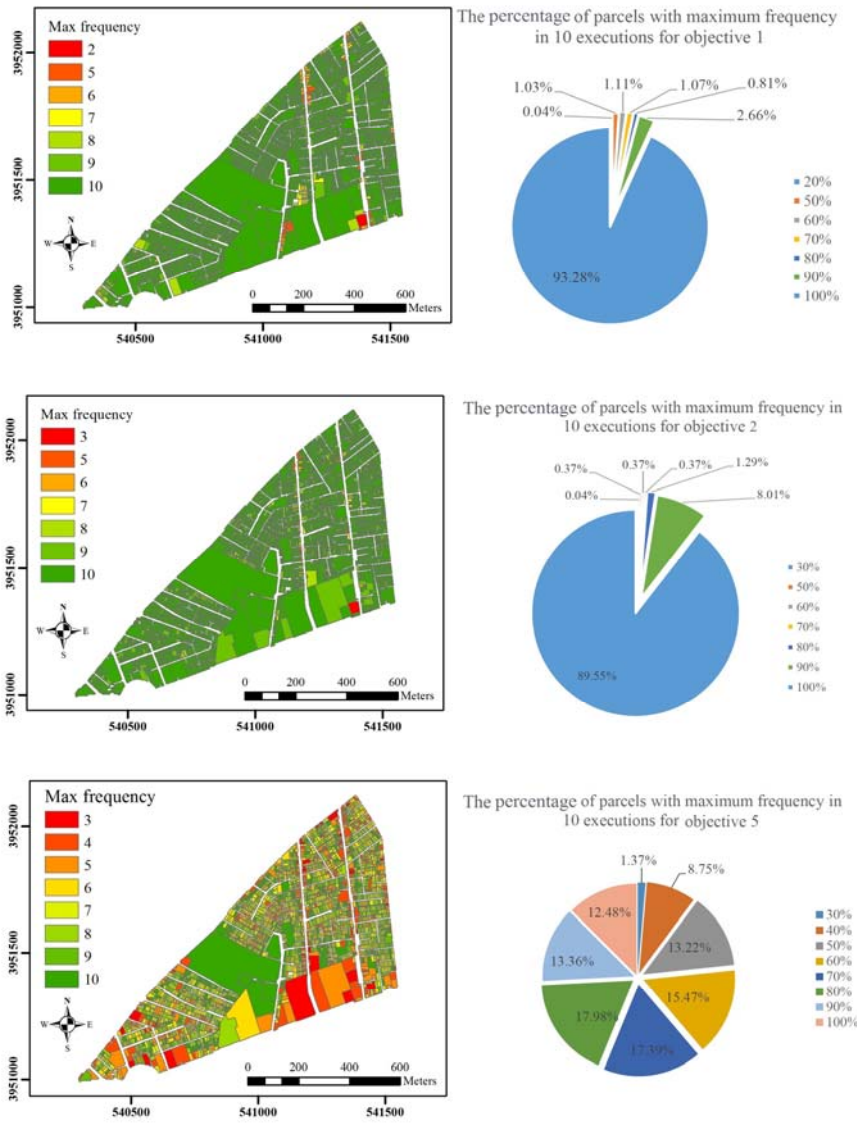


Figure 9. Overlap percentage of land-uses in different executions

Table 4 also indicates that objective 5 has more changes in objective value range over the ten runs, while the worst and best values in the first four objectives are very close together.

Table 4. The worst and the best objective values in different executions

	objective 1		**objective 2**		**objective 3**		**objective 4**		**objective 5**	
	Worst	Best	Worst	Best	Worst	Best	Worst	Best	Worst	Best
Run1	0.1038	0.1969	0.0388	0.0727	0.1417	0.5337	0.2505	0.7599	9.4177	0
Run2	0.1033	0.1971	0.0402	0.0712	0.1483	0.5337	0.2488	0.76	9.4039	0
Run3	0.1036	0.1974	0.0387	0.0727	0.1418	0.5341	0.25	0.7601	10.0576	0
Run4	0.1037	0.1973	0.0399	0.0727	0.1414	0.5337	0.2535	0.7601	9.9615	0
Run5	0.1035	0.1973	0.0391	0.0727	0.1425	0.5337	0.2578	0.7602	9.621	0
Run6	0.1037	0.1974	0.0396	0.0727	0.1417	0.534	0.253	0.7602	10.3944	0
Run7	0.1032	0.1974	0.0419	0.0726	0.1407	0.534	0.2532	0.7599	9.4346	0
Run8	0.1032	0.1971	0.0376	0.0724	0.1451	0.5341	0.2467	0.7599	10.2337	0
Run9	0.1044	0.1974	0.041	0.0727	0.1414	0.5341	0.244	0.7599	9.386	0
Run10	0.104	0.1972	0.0393	0.0726	0.1414	0.5337	0.2536	0.76	9.3485	0

Among all PF solutions, five solutions are important. These solutions show the best land-use layout for each objective. Figure 10 illustrates the five land-use layouts. As shown by Figure 10 (a), in the best layout for objective 1 (compatibility), the largest area is allocated to residential, park, and commercial land-uses. Theses land-use types have more compatibility with each other, and joining them means that the overall land-use compatibility of the study area can be maximized.

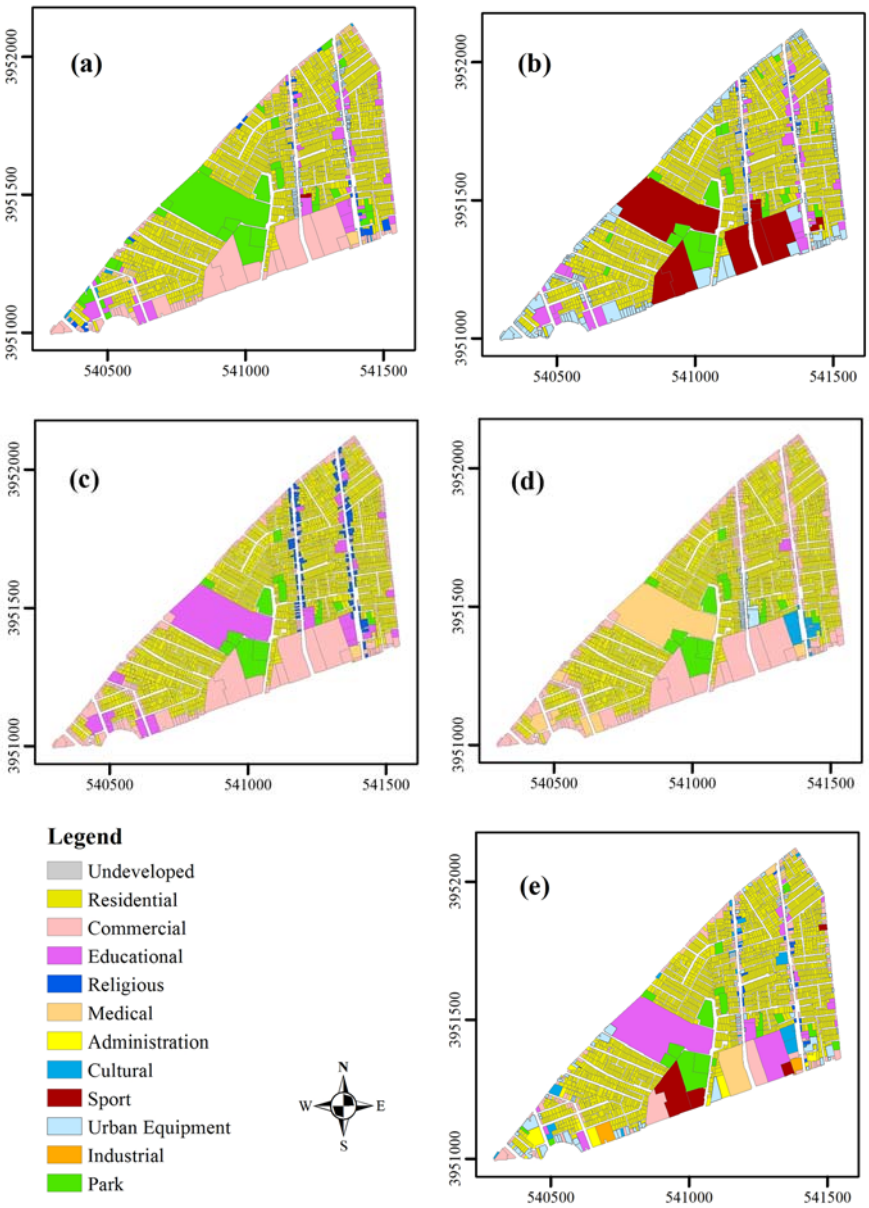


Figure 10. The best land-use layout for objective 1 (a), objective 2 (b), objective 3 (c), objective 4 (d) and objective 5 (e)

Figure 10 (b) shows the best land-use layout for achieving maximum dependency between land-units. In this layout, except for residential land-use that prevails in all layouts, sport and urban equipment land-uses are dominant. These land-use types are heavily dependent on residential land-use, and their presence in the layout increases the dependency between land-units. Figure 10 (c) shows the best arrangement with respect to physical suitability. In this arrangement, commercial, educational and religious land-use types are allocated to most land-units. According to the parameters of physical suitability [43], these land-use types maximize the physical suitability of overall arrangement. Figure 10 (d) shows the best arrangement with respect to compactness. The same land-use types in different parts of the map are located next to each other. Moreover, in this

layout, the variety of land-use types is lower than that of other layouts. The reason for this is to achieve maximum compactness between neighboring land-uses. Finally, Figure 10 (e) indicates the best arrangement of land-uses with respect to per capita demand. As shown in this layout, the variety of land-uses is more pronounced than in other layouts. According to the high per capita demand of educational land-use, the algorithm devotes the largest parcel of the study area to this land-use type. It is notable to say that the arrangements in Figure 10 belong to the extreme values of each objective function (the edge of PFs) in which just one objective function is optimized and other objectives are not considered. We just want to show that the decision-makers can see different arrangements of land-use types and they can select one of them as the desired arrangement. However, urban planners usually seek the balanced status of objective functions.

6. Evaluation and discussion

The results of NSGA-II and NSGA-III in urban land-use planning are used to evaluate the results of the proposed algorithm because they are in the same family of EMOs. In these two algorithms, the EMUP operator is also employed to achieve comparable results with respect to HNSGAIII because the EMUP significantly affects the diversity of solutions. Moreover, all initial values have been considered the same to compare the results fairly. As mentioned before, in NSGA-III, extreme points may be duplicated during the normalization step; in this case, the infinite hyper-plane can be defined to include these extreme points [31]. To avoid such a problem, the normalization method described in [31] is employed during the implementation of NSGA-III.

Moreover, to be precise, two more real instances were examined in this section. The land-use maps of the different parts of two other cities at the parcel level on the scale 1:2000 were used to implement and test the proposed algorithm. These case studies areas, similar to the main case study area adopted in this work, are located in deteriorated urban environments, and the renovation of buildings and the change of land-uses are priorities for the municipalities. The presence of various land-use types at different service levels is one of the features of these two case study areas. The numbers of urban parcels in these study areas are 2238 and 2685, respectively. Figure 11 displays the urban parcels and their main land-uses types in these two study areas.

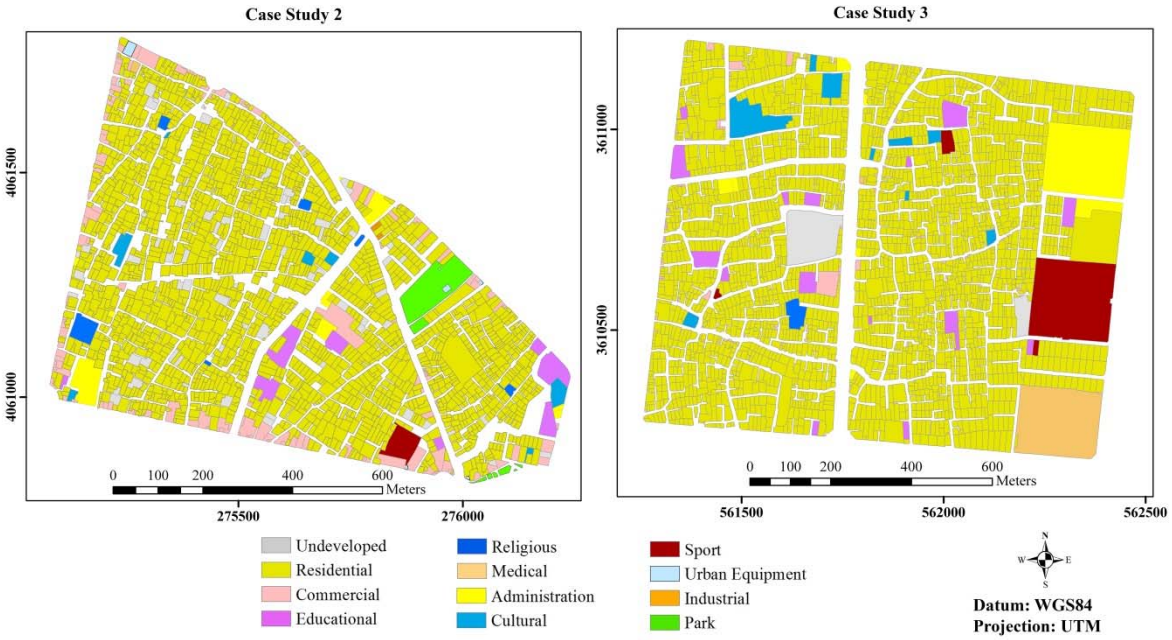


Figure 11. Three case study areas and the main land-use classes.

A large number of performance measures have been proposed for comparing the results of multi- and many-objective optimization methods [50, 56]. Given that the true PF is not achievable in this study, performance indicators that do not require the true PF should be used. Therefore, the

Diversity Comparison Indicator (DCI) [56] and the HV indicator [51] were selected to compare the results of the proposed algorithm with respect to the results of two other algorithms in three case study areas, due to their independence from the true PF.

The DCI is designed to compare the diversity of different PF approximations. This indicator calculates the diversity of the solutions in different PF approximations by dividing the objective space by the number of cells. The DCI value of a PF approximation is a number in the range [0, 1], and a greater value indicates better diversity. The number of divisions (div) in objective space is an important parameter for this indicator Li, Yang [56]. The DCIs of the results are calculated according to different div values and are shown in Figure 12 in three case study areas. As shown in this figure, in all case studies, the DCI values of the proposed algorithm are greater than those of the other two algorithms in almost all divisions, which indicates that the proposed algorithm works better in terms of diversity. By increasing div, the DCI is decreased for all three algorithms. This is due to the fact that the number of non-empty cells in objective space increases with div, and for a solution set, it is essential to consider more cells which are filled by other solution sets [56]. Additionally, in the MOLUP, DCI indicates that NSGA-II performs better than NSGA-III in terms of diversity.

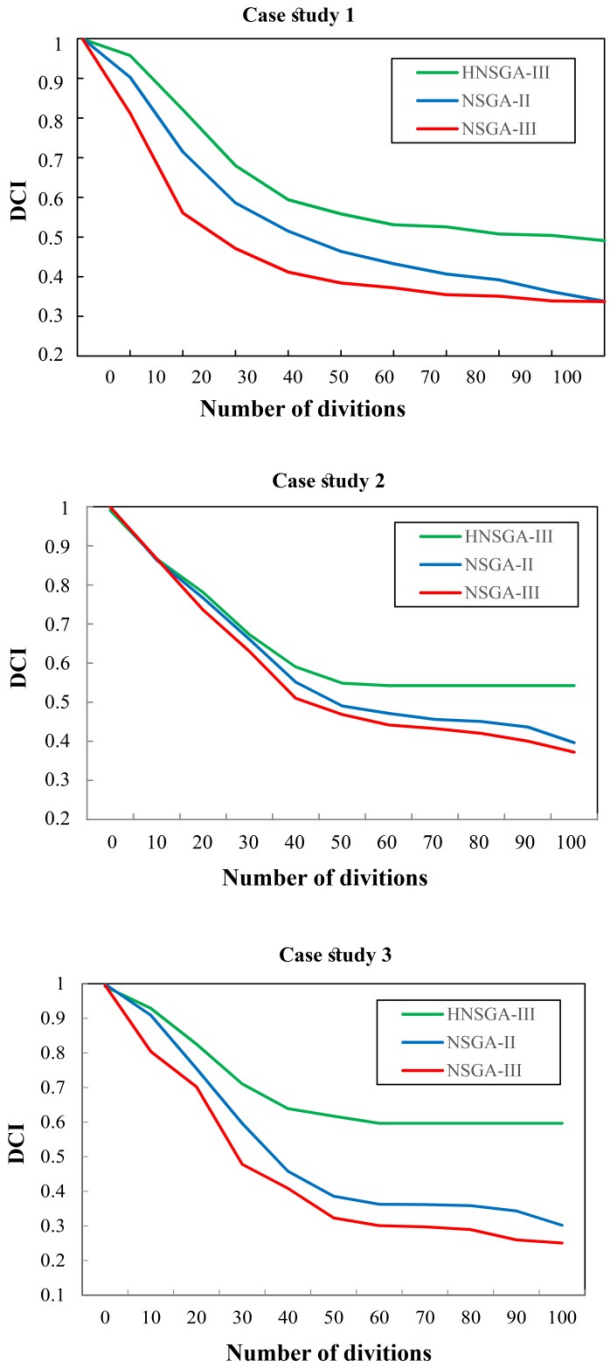


Figure 12. DCI values of the NSGAII, NSGA-III and the proposed algorithm next to different numbers of divisions in three case study areas.

Figure 13 shows variations in the average HV of ten runs for the three algorithms compared for the three case study areas. This figure shows that all three algorithms are able to increase the HV value when increasing the number of iterations. The proposed algorithm manages to find a better HV value than the other two algorithms, even after 50 iterations, while the NSGA-III does not find a better HV value than any of the other two algorithms. This is the case for all three case study areas.

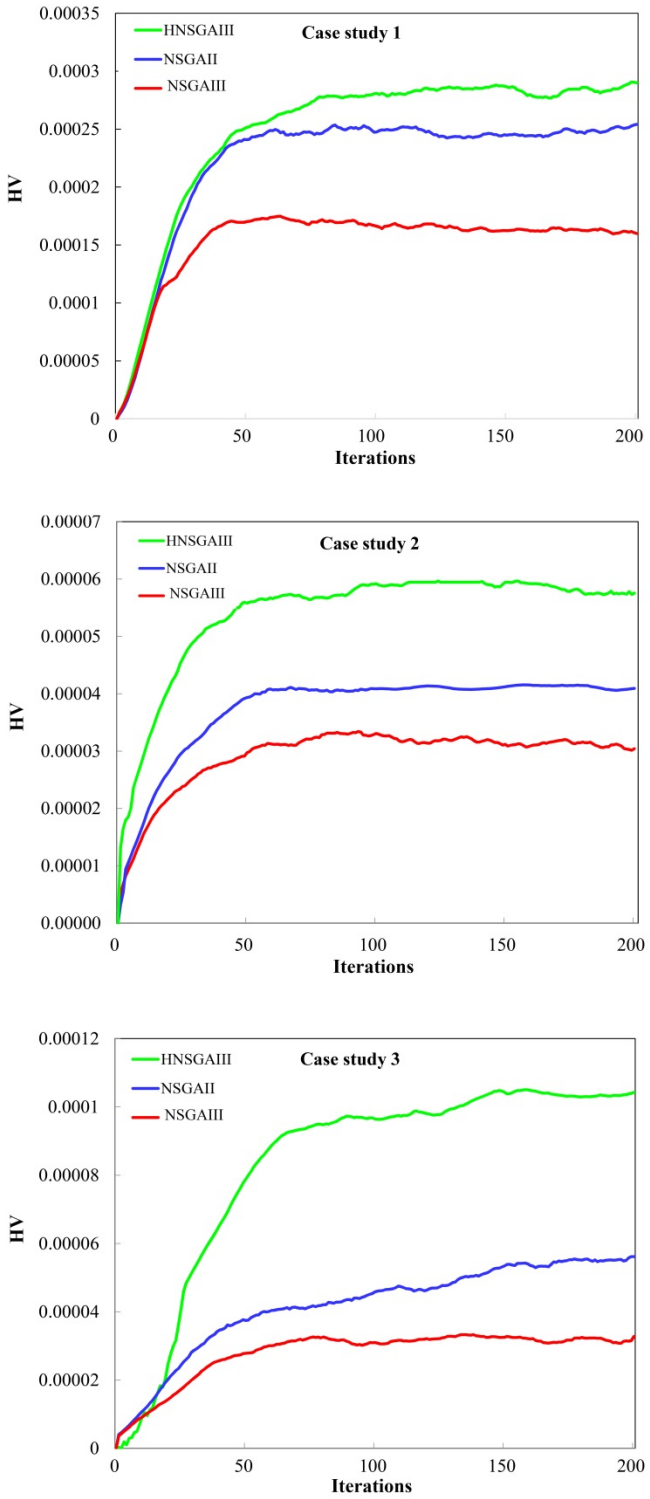


Figure 13. Variation of the average HV indicator of ten runs with respect to the number of iterations.

Moreover, Vargha-Delaney A12 effect size values [57] have been used to quantify the outperformance difference of the algorithms. We computed the effect size values of two comparison metrics i.e. HV and DCI. Table 5 presents the effect size values of mean HV and mean DCI for the comparison. According to [57] values in [0.64, 0.71] indicates medium differences and values in [0.71, 1] indicate large differences. Therefore, effect size values show that the differences between the comparison parameters of the proposed algorithm with the other two algorithms are statistically large and the HV and DCI values reported in the proposed algorithm are better.

Table 5. Effect size values (A12) for comparison HV and DCI indicator in 3 employed algorithms

Algorithms		Effect size of HV	Effect size of DCI
HNSGAIII	NSGAII	0.7223	0.6942
HNSGAIII	NSGAIII	0.8974	0.7934
NSGAII	NSGAIII	0.8851	0.6446

Figure 14 presents the parallel coordinates plot for 3 case study areas. In this diagram, the objective functions are normalized between 0 to 1. As seen in this figure, the spread of solutions over solution space is more regular in the proposed algorithm in comparison with NSGA-III and NSGA-II in all case studies. To investigate more, Figure 15 shows the PFs of two combinations of objectives in three dimensions for each algorithm in case study 1. Of course, parallel coordinate plots show all 5 objective functions simultaneously but three-dimensional graphs in Figure 15 are provided to a better understanding of the scatter of the solution in the PFs. According to these figures, the proposed algorithm appears to perform better than the other two algorithms in generating diverse solutions. Moreover, in this figure, the solutions of NSGA-II have more diversity than the solutions of NSGA-III.

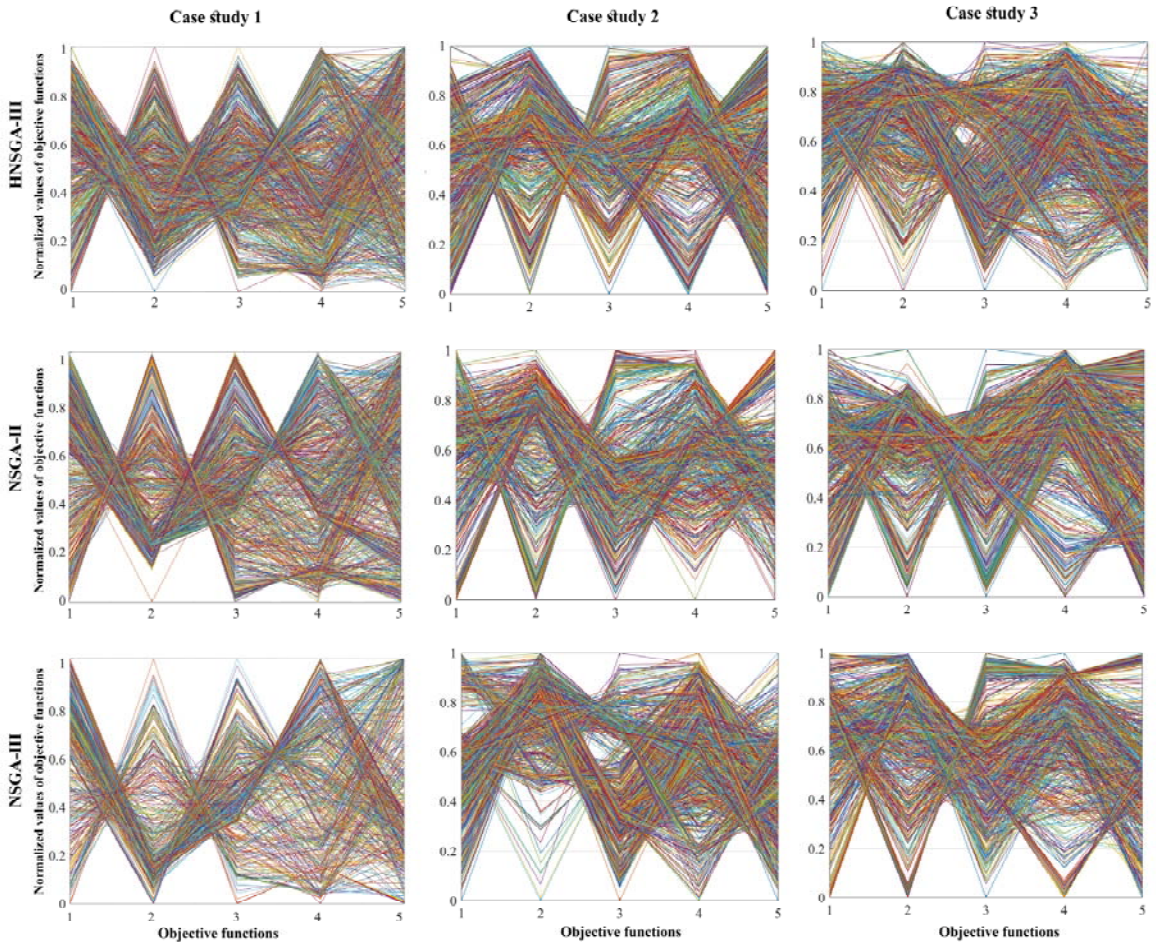


Figure 14. Parallel coordinates plot of the objective functions for the 3 case study areas

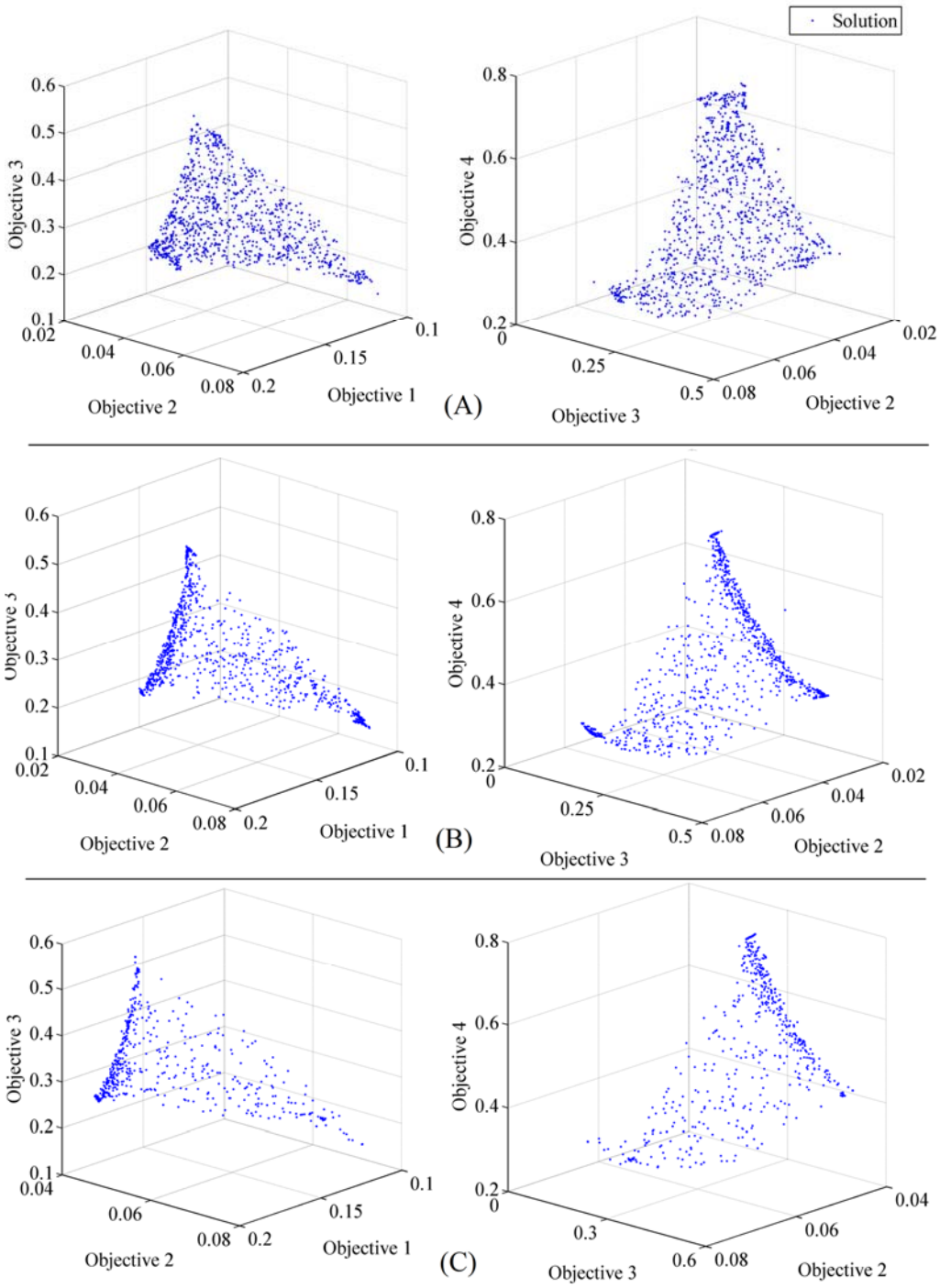


Figure 15. The 3D PF of three algorithms; (A). HNSGA-III, (B). NSGA-II, and (C). NSGA-III

Consequently, as Figures 14 and 15 demonstrate, using hypercubes improves the spread of the solutions and generates diverse solutions. However, one of the problems with hypercubes is determining their optimal number. Low numbers cause the lack of proper diversity of solutions, and high numbers impose additional calculations on the program.

Since the final optimal solutions can never be reached in NP-hard problems (like the problem of this study), a threshold should be considered for comparing different algorithms' results. HV as a suitable indicator is used to define this threshold. Given that the HV indicator includes both the amount of convergence (number of iterations) and the solution spread (initial population), it is a good criterion for comparing different algorithms. We can assess the HV values calculated by the algorithms in terms of time, using two approaches. In the first approach, the HV values are recorded in different time steps by the three algorithms and are shown in Figure 16(a) in case study 1. As the

figure demonstrates, the proposed algorithm at each time step produces better results in comparison with the other two algorithms. In the second approach, the time needed to reach a certain value of HV for each algorithm is recorded and is shown in Figure 16(b). In this figure, it is also clear that the proposed algorithm reaches a certain amount of HV in less time than the two other algorithms. Figure 16(b) shows that the NSGA-III never reaches HV = 0.00018 and the NSGA-II never reaches HV = 0.00027. As a result, we can say that, when the same hardware and software system and identical conditions are applied, the proposed algorithm generates a higher HV value in less time than the two other algorithms. As mentioned before, the EMUP is considered for all three algorithms in this comparison. Although this operator finds better solutions and expands the solution space, local search and the related calculations to the implementation of this operator slow down the execution of the algorithms.

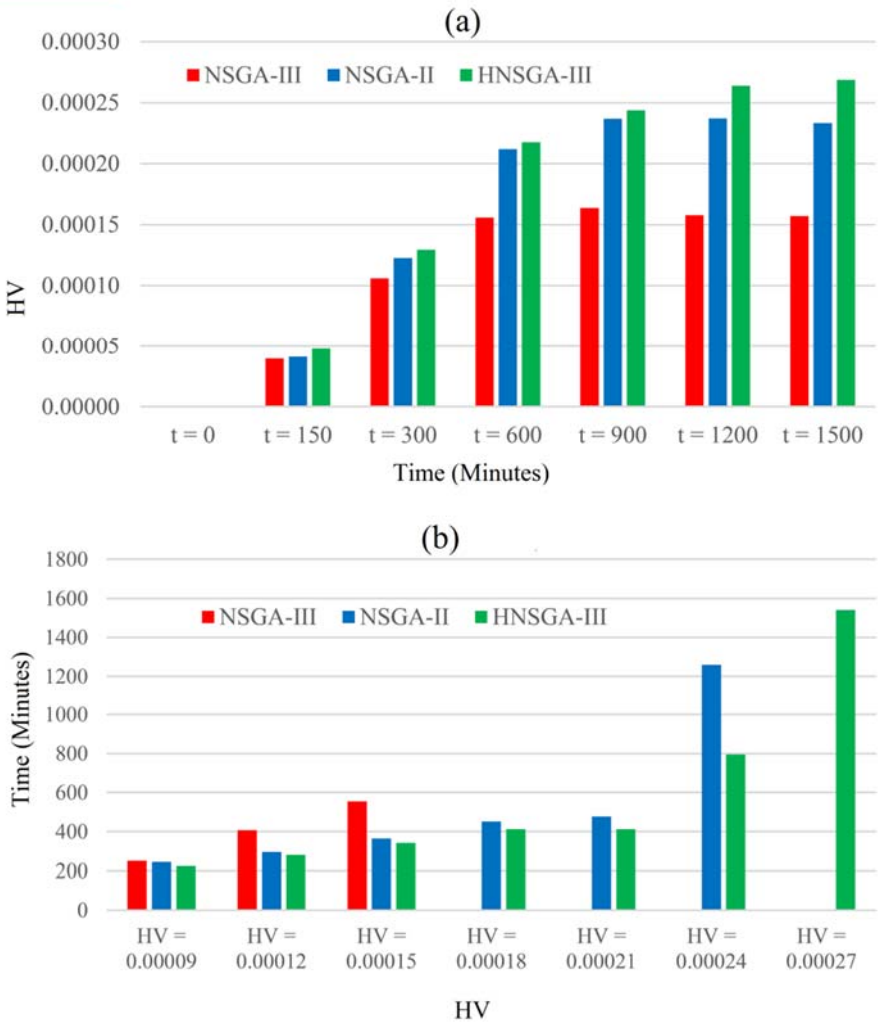


Figure 16. Comparison of computational times using the HV indicator: (a) average HV indicator values at different time steps; (b) average time at different HV values.

Moreover, the comparison of this work and the research of Masoumi, Coello [20] which employed the original NSGA-II, showed that the use of hypercubes and EMUP spread the solution space significantly which leads to better solutions (with better objective function values). However, the objective functions of this research are different from those adopted by Masoumi, Coello [20]. In that previous work, per-capita demand was considered as a criterion but in this paper, per-capita violation is considered as an objective function to be minimized during the optimization process.

Discussing the validity of these results in the real world is an issue that needs to be addressed. For this purpose, in this research, the results of previous researchers in this field as well as a survey of 7

urban design experts have been used. In a general summary of the opinions of experts, it can be said that the results of this research can be useful to urban design and management decision-makers in the following cases.

- In presenting the initial plans in the detailed design of cities, it meets the needs of urban users for accurate calculations of important objective functions such as compatibility, dependency, suitability, compactness, and per capita demand. The current routine is that designers in designing new cities or detailed plans, produce original plans according to their experiences and modify them based on urban design rules, and there is no codified calculation procedure for this. This can be especially useful in smart cities where the whole design process is based on complex initial calculations.

- The effects of changing a land-use on the existing layout of urban uses are fully computable. This matter can help the decision-maker to easily decide whether or not to change. Relevant calculations are not very common in traditional systems and the effects of change up to a very limited effect radius have been extracted qualitatively.

There is a gap between the results of this research and its implementation in the real world, like other applications of new algorithms. The most important thing in reducing this gap is to model real-world problems and incorporate them into the algorithm. For example, social and economic issues, although hidden in the first two objective functions of this research, but if they have been considered as separate objective functions, the results will be closer to the real world. Another issue is the management of property owners, which should consider strategies for convincing them, which is beyond the scope of this article and can be a new field of work. However, this is a very new topic in urban design and management and requires a lot of work to get closer to the real-world application.

7. Conclusion

Urban land-use planning as a many-objective optimization problem requires finding all non-dominated optimum land-use arrangements. In this research, NSGA-III is improved based on and the use of hypercubes for its use in the MOLOUP problem. Given the nature of ULUP problem, the convergence of solutions is slow and a lot of trial and error is required if they are to reach their optimal state. As such, in this study, a new mutation operator called EMUP is introduced. The results show that the proposed algorithm converges in a few steps to the PF with this operator.

The five objective functions of this research have always been considered by urban designers in designing new cities and also proposing detailed plans, but scientific and simultaneous calculations of these functions together are a new issue that allows urban decision-makers to be able to consider different views of the city. On the other hand, solving this problem by using multi-objective algorithms gives a very scientific and suitable opportunity to urban affairs decision-makers to be able to see different aspects of the city by considering different objective functions. But solving ULUP problem using many-objective algorithms has many complexities in data type, objective functions, constraints, and search space. Therefore, using new algorithms in this problem requires a lot of adaptation, and the possibility of evaluating the algorithm using the usual evaluation parameters in many-objective calculations is very difficult or sometimes impossible.

One of the limitations of this study is the large number of solutions in the PF. As part of our future work, we aim to propose a model designed to select a solution that all the stakeholders in the process of urban land-use planning agree upon, after being presented every solution. This model can use the modelling techniques of interactions and negotiation between stakeholders, such as interactive and agent-based models, to aggregate the preferences of the stakeholders and select a solution that has the most compatibility with these preferences.

According to these explanations and experiences of this research and other researchers in this field, achieving the applicable form of many-objective algorithms in ULUP requires the

implementation and realization of many parameters in the real world, which can be a new way for future research.

8. References

1. Ligmann-Zielinska, A., R.L. Church, and P. Jankowski, *Spatial optimization as a generative technique for sustainable multiobjective land-use allocation.* International Journal of Geographical Information Science, 2008. **22**(6): p. 601-622.
2. Li, X. and L. Parrott, *An improved Genetic Algorithm for spatial optimization of multi-objective and multi-site land use allocation.* Computers, Environment and Urban Systems, 2016. **59**: p. 184-194.
3. Chuvieco, E., *Integration of linear programming and GIS for land-use modelling.* International Journal of Geographical Information Science, 1993. **7**(1): p. 71-83.
4. Moah, H. and P. Kanaroglou, *A tool for evaluating urban sustainability via integrated transportation and land use simulation models.* Urban Environment, 2009. **3**: p. 28–46.
5. Khalili-Damghani, K., et al., *Solving land-use suitability analysis and planning problem by a hybrid meta-heuristic algorithm.* International Journal of Geographical Information Science, 2014.
6. Porta, J., et al., *High performance genetic algorithm for land use planning.* Computers, Environment and Urban Systems, 2013. **37**: p. 45-58.
7. Koc, I. and I. Babaoglu, *A comparative study of swarm intelligence and evolutionary algorithms on urban land readjustment problem.* Applied Soft Computing, 2020: p. 106753.
8. Balling, R.J., et al., *Multiobjective urban planning using genetic algorithm.* Journal of Urban Planning and Development, 1999. **125**(2): p. 86-99.
9. Masoomi, Z., *Modeling the physical effects of urban land-uses change using optimization algorithms and spatial analysis (In persian)*, in *Geodesy and Geomatics engineering, Geospatial Information Systems (GIS)*. 2013, Khajeh Nasir Toosi University Of Technology.
10. Stewart, T.J., R. Janssen, and M. van Herwijnen, *A genetic algorithm approach to multiobjective land use planning.* Computers & Operations Research, 2004. **31**(14): p. 2293-2313.
11. Aerts, J.C. and G.B. Heuvelink, *Using simulated annealing for resource allocation.* International Journal of Geographical Information Science, 2002. **16**: p. 571–587.
12. Ai, B., S. Ma, and S. Wang, *Land-use zoning in fast developing coastal area with ACO model for scenario decision-making.* Geo-spatial Information Science, 2015. **18**(1): p. 43-55.
13. Shifa, M., et al., *Land-use spatial optimization based on PSO algorithm.* Geo-Spatial Information Science, 2011. **14(1)**: p. 54–61.
14. Yang, L., et al., *An improved artificial bee colony algorithmfor optimal land-use allocation.* International Journal of Geographical Information Science, 2015: p. 1-20.
15. Mohammadi, M., M. Nastaran, and A. Sahebgharani, *Development, application, and comparison of hybrid meta-heuristics for urban land-use allocation optimization: Tabu search, genetic, GRASP, and simulated annealing algorithms.* Computers, Environment and Urban Systems, 2016. **60**: p. 23-36.
16. Romaniello, M. and U. Fiore, *Special issue on recent advances in soft set decision making: Theories and applications.* Applied Soft Computing, 2017. **100**(54): p. 364-365.
17. Cao, K., et al., *Spatial multi-objective land use optimization: extensions to the non-dominated sorting genetic algorithm-II.* International Journal of Geographical Information Science, 2011. **25**(12): p. 1949-1969.

18. Huang, K., et al., *An improved artificial immune system for seeking the Pareto front of land-use allocation problem in large areas.* International Journal of Geographical Information Science, 2013. **27**(5): p. 922-946.
19. Feng, C.M. and J.J. Lin, *Using a genetic algorithm to generate alternative sketch maps for urban planning.* Computers Environment and Urban Systems, 1999. **23**: p. 91–108.
20. Masoumi, Z., C.A.C. Coello, and A. Mansourian, *Dynamic urban land-use change management using multi-objective evolutionary algorithms.* Soft Computing, 2020. **24**(6): p. 4165-4190.
21. Deb, K., et al., *A fast and elitist multiobjective genetic algorithm: NSGA-II.* Evolutionary Computation, IEEE Transactions on, 2002. **6**(2): p. 182-197.
22. Shaygan, M., et al., *Spatial Multi-Objective Optimization Approach for Land Use Allocation Using NSGA-II.* IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING, 2014.
23. Heyns, A.M. and J.H. van Vuuren, *A multi-resolution approach towards point-based multi-objective geospatial facility location.* Computers, Environment and Urban Systems, 2016. **57**: p. 80-92.
24. Jansen, T., *Analyzing evolutionary algorithms: The computer science perspective*. 2013: Springer Science & Business Media.
25. Deb, K., *Recent advances in evolutionary multi-criterion optimization (EMO)*, in *Proceedings of the Genetic and Evolutionary Computation Conference Companion*. 2017, ACM: Berlin, Germany. p. 702-735.
26. Köppen, M. and K. Yoshida. *Substitute distance assignments in NSGA-II for handling many-objective optimization problems*. in *Evolutionary Multi-Criterion Optimization*. 2007. Springer.
27. Masoumi, Z., et al., *Using an Evolutionary Algorithm in Multiobjective Geographic Analysis for Land Use Allocation and Decision Supporting.* Geographical Analysis, 2017.
28. Deb, K. and H. Jain, *An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: solving problems with box constraints.* Evolutionary Computation, IEEE Transactions on, 2014. **18**(4): p. 577-601.
29. Das, I. and J.E. Dennis, *Normal-boundary intersection: A new method for generating the Pareto surface in nonlinear multicriteria optimization problems.* SIAM Journal on Optimization, 1998. **8**(3): p. 631-657.
30. Chiang, T., *nsga3cpp: A C++ implementation of NSGA-iii*. 2014.
31. Yuan, Y., H. Xu, and B. Wang, *An improved NSGA-III procedure for evolutionary many-objective optimization*, in *Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation*. 2014, ACM: Vancouver, BC, Canada. p. 661-668.
32. Ishibuchi, H., et al. *Performance comparison of NSGA-II and NSGA-III on various many-objective test problems*. in *2016 IEEE Congress on Evolutionary Computation (CEC)*. 2016.
33. Taleai, M., et al., *Evaluating the compatibility of multi-functional and intensive urban land uses.* International Journal of Applied Earth Observation and Geoinformation, 2007. **9**(4): p. 375-391.
34. Cao, K., W. Zhang, and T. Wang, *Spatio-temporal land use multi-objective optimization: A case study in Central China.* Transactions in GIS, 2019. **23**(4): p. 726-744.
35. Schwaab, J., et al., *Improving the performance of genetic algorithms for land-use allocation problems.* International Journal of Geographical Information Science, 2018. **32**(5): p. 907-930.

36. Handayanto, R.T., et al., *Achieving a sustainable urban form through land use optimisation: insights from Bekasi City's land-use plan (2010–2030).* Sustainability, 2017. **9**(2): p. 221.
37. Song, M. and D. Chen, *An improved knowledge-informed NSGA-II for multi-objective land allocation (MOLA).* Geo-spatial Information Science, 2018. **21**(4): p. 273-287.
38. Cao, K., et al., *Sustainable land use optimization using Boundary-based Fast Genetic Algorithm.* Computers, Environment and Urban Systems, 2012. **36**(3): p. 257-269.
39. Abolhasani, S., et al., *Simulating urban growth under planning policies through parcel-based cellular automata (ParCA) model.* International Journal of Geographical Information Science, 2016: p. 1-26.
40. Tong, Z., et al., *Quantification of the openness of urban external space through urban section.* Geo-spatial Information Science, 2020. **23**(4): p. 316-326.
41. Masoumi, Z., J. van L Genderen, and J. Maleki, *Fire Risk Assessment in Dense Urban Areas Using Information Fusion Techniques.* ISPRS International Journal of Geo-Information, 2019. **8**(12): p. 579.
42. Maleki, J., F. Hakimpour, and Z. Masoumi, *A Parcel-Level Model for Ranking and Allocating Urban Land-Uses.* ISPRS International Journal of Geo-Information, 2017. **6**(9): p. 273.
43. Masoomi, Z., M.S. Mesgari, and M. Hamrah, *Allocation of urban land uses by Multi-Objective Particle Swarm Optimization algorithm.* International Journal of Geographical Information Science, 2013. **27**(3): p. 542-566.
44. Mezura-Montes, E. and C.A.C. Coello, *Constraint-handling in nature-inspired numerical optimization: past, present and future.* Swarm and Evolutionary Computation, 2011. **1**(4): p. 173-194.
45. Maab-Consulting-Engineers, *Definitions and concepts of urban land-uses and determining the per capita (In Persian)*. 2010: Iran's Supreme Council for Planning and Architecture.
46. Shao, Z., et al., *Urban sprawl and its impact on sustainable urban development: a combination of remote sensing and social media data.* Geo-spatial Information Science, 2020: p. 1-15.
47. Chankong, V. and Y.Y. Haimes, *Multiobjective Decision Making Theory and Methodology*. 1983: New York: North-Holland.
48. Farnahad, C.E., *Development pattern of district 7 of Tehran (In Persian)*. 2005: Orderd by Tehran municipality.
49. Habibi, M. and S. Masaeli, *Urban land uses per capita (In Persian)*. 1999, Tehran: National Land and Housing Organization.
50. Riquelme, N., V. Von Lücken, and B. Baran. *Performance metrics in multi-objective optimization*. in *2015 Latin American Computing Conference (CLEI)*. 2015.
51. Bradstreet, L., *The hypervolume indicator for multi-objective optimisation: calculation and use*. 2011: University of Western Australia.
52. Cao, Y., B.J. Smucker, and T.J. Robinson, *On using the hypervolume indicator to compare Pareto fronts: Applications to multi-criteria optimal experimental design.* Journal of Statistical Planning and Inference, 2015. **160**: p. 60-74.
53. Coello, C.A.C., G.B. Lamont, and D.A. Van Veldhuizen, *Evolutionary algorithms for solving multi-objective problems*. Vol. 5. 2007: Springer.
54. Bajestani, M.A., et al., *A multi-objective scatter search for a dynamic cell formation problem.* Computers & operations research, 2009. **36**(3): p. 777-794.

55. Li, X.-y., J.-h. Zheng, and J. Xue, *A Diversity Metric for Multi-objective Evolutionary Algorithms*, in *Advances in Natural Computation: First International Conference, ICNC 2005, Changsha, China, August 27-29, 2005, Proceedings, Part III*, L. Wang, K. Chen, and Y.S. Ong, Editors. 2005, Springer Berlin Heidelberg: Berlin, Heidelberg. p. 68-73.
56. Li, M., S. Yang, and X. Liu, *Diversity comparison of Pareto front approximations in many-objective optimization.* IEEE Transactions on Cybernetics, 2014. **44**(12): p. 2568-2584.
57. Vargha, A. and H.D. Delaney, *A critique and improvement of the CL common language effect size statistics of McGraw and Wong.* Journal of Educational and Behavioral Statistics, 2000. **25**(2): p. 101-132.