Expert Systems With Applications 188 (2022) 116082

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

# A novel metaheuristic algorithm by efficient crossover operator for land readjustment

Ismail Koc [a,*], Tayfun Cay [b], Ismail Babaoglu [c]

[a] Department of Software Engineering, Konya Technical University, 42075 Konya, Turkey
[b] Department of Computer Engineering, Konya Technical University, 42075 Konya, Turkey
[c] Department of Geomatics Engineering, Konya Technical University, 42075 Konya, Turkey

ARTICLE INFO

*Keywords:*
Subdivision
Land Readjustment
Land Reallocation
Zoning Application
Metaheuristic Algorithms
Intelligent Parcel-Based Crossover
Expert System

ABSTRACT

Land readjustment and reallocation (LR) applications are complex and difficult real-world problems involving many different criteria. By considering these criteria, it is very difficult and takes a long time to be solved manually by an expert. Since the search space of these problems is very large, solution of these problems requires *meta*-heuristic optimization algorithms instead of classical methods in order to acquire more robust, acceptable and qualified solutions. Considering the *meta*-heuristic approaches, the algorithm needs an objective function that can make the right decision and evaluate the solutions most reasonably among the candidate solutions. Using the proposed objective function, the quality of the distribution and subdivision plans will be automatically evaluated and compared without the need for an expert. In this study, an objective function which considers all the criteria in the LR problems is proposed. In addition, unlike the available crossover operators used in *meta*-heuristic algorithms in the literature, two different parcel-based crossover operators called Classical (CPC) and Intelligent (IPC) Parcel-Based Crossover Operators are proposed. While CPC performs the distribution of the owners to the predetermined parcel randomly, IPC makes this operation with a greedy approach rather than randomly. According to this approach, if the shareholder and distance values after the crossover operation would be better than the existing ones, the crossover operation is performed. Otherwise, this operation is cancelled. By using the proposed objective function and crossover operators, artificial bee colony (ABC), particle swarm optimization (PSO) and differential evolution (DE) algorithms are run under equal conditions on a real project site, and the obtained results are compared with the official results obtained by a technician in the study. In addition, since there will be so many zoning blocks of different sizes and shapes on a real project site, it is very possible to have gaps or overflows in the blocks of subdivision plans obtained from the algorithms. Therefore, the gaps and overflow areas in the blocks can be completely eliminated by utilizing an Expert System developed specifically for LR problems called LRES, and as a result, the solutions obtained from the algorithms can be directly applicable in real life by the LRES. It's clearly seen from the experimental studies that all of the results obtained by using the algorithms based on LRES are much more effective than the official results obtained by a technician in terms of both solution quality and speed. In addition, among the evaluated algorithms, it is observed that the PSO algorithm presents much more effective and robust results than results of the other algorithms. Moreover, as a consequence of the algorithms using the IPC presents much more successful results than the results of the algorithms using CPC, it can be used as a very effective alternative crossover operator for land use problems.

## 1. Introduction

Due to the limited amount of available and usable land, the urban land should be used in a planned and optimal way in order to be able to meet the socioeconomic and cultural needs that change over time depending on population increase. The basis of a healthy and livable urbanization depends on urban practices. Urban areas are developing due to socio-economic and socio-cultural reasons, and their population are growing, and need new settlement areas or improvement of the existing settlement. Urban transformation practices, on the other hand,

* Corresponding author.
*E-mail addresses:* ismailkoc@ktun.edu.tr (I. Koc), tcay@ktun.edu.tr (T. Cay), ibabaoglu@ktun.edu.tr (I. Babaoglu).

https://doi.org/10.1016/j.eswa.2021.116082
Received 28 May 2020; Received in revised form 27 July 2021; Accepted 11 October 2021
Available online 15 October 2021
0957-4174/

offer a solution for these needs by reproducing the existing residential areas.

Migration and squatter settlements in Turkey especially in the post-1950 have adversely affected the production of urban space. This situation led to the formation of stratification with distorted urbanization, unhealthy conditions and sociocultural structure problems. At the same time, this situation has caused the lands to become plot by urban sprawl. Accordingly, land and plot arrangement applications have come up with the start of the planned development works and 3194 Zoning Law especially in Turkey, and a planned development period has started in all local governments (İnam, 2013).

Land readjustment (LR) regulates the appropriate social, cultural and technical infrastructures necessary to provide living areas for urban communities (Colak & Memisoglu, 2019). LR plans are development application tools that transforms unusable cadastral parcels into a more economically usable structure for generating an effective future planning of city. In other words, LR is described as an effective technique converting a useless land parcel into an economically usable urban parcel (Chou & Shen, 1982; Doebele, 1982; Yomralioglu, 1992). The most important features of LR technique are its practical use in very large areas and to minimize the time of the plan implementation. The LR presents some advantages for both land owners and public. Thanks to LR applications, public areas such as roads, squares, parks, car parks, children's playgrounds, green areas, police stations and primary and secondary education areas are created. The LR also provides higher standard and stable land values and minimizes property sharing (Türk, 2009).

When the available literature in Turkey researches investigated, the studies generally addresses on the actualization of the techniques in terms of financing and administration. However, there are very few studies that focus on technical details of subdivision and distribution (reallocation) in the literature (Kucukmehmetoglu & Geymen, 2016). In his study, Yomralioglu (1992) focuses on the issue of evaluation during the redistribution of parcels, and examined what factors might affect the value of a parcel in land readjustment (LR) studies and how these factors should be handled. The author proposed a new solution for the distribution of parcels in its studies in LR. Uzun (1992) studies on different zoning parcel production methods such as producing standard size parcels, parcel production by allocation areas and parcel production by number of shareholders in urban area arrangements. Turk (2005) aims to develop new lands and reorganize the places built in urban areas, as well as to provide the production of urban areas and areas required for public services. Turk (2007) defines the basic conditions for successful and efficient application within the framework of international literature. Yomralioglu, Nisanci, and Yildirim (2007) carry out an application for Trabzon province regarding nominal asset based Land Arrangement with the help of raster based land assessment map. Kokturk and Kokturk (2009) offers equivalence criteria with minimum deviation principle before and after LR. Uzun (2009) provides information on application methods of urban land development plans in Turkey. Çete (2010) discusses importance of LR in the implementation of development plans in Turkey. Uygun (2011) utilizes fuzzy logic method for solving redistribution problem according to value bases criteria in zoning applications. Yilmaz, Çağdaş, and Demir (2015) examine a wide range of articles on land arrangement to create a framework and methodology to help evaluate and compare national LR processes. Kucukmehmetoglu and Geymen (2016) carry out subdivision and distribution operations using Integer Programming and Linear Programming techniques for urban area arrangement. Çağdaş and Linke (2020) aim to contribute to the current study by giving an analysis for the case study in Turkey based on the Institutional Analysis and Development Framework. Coruhlu, Uzun, and Yildiz (2020) aim to develop recommendations to eliminate the problem of legal confiscation in the light of the European Court of Human Rights decisions. Koc, Cay, and Babaoglu (2020) present new approaches for automated land subdivision using binary search algorithm in zoning applications. Koc and Babaoglu (2021) implement five *meta*-heuristic algorithms to LR problems, and employ a novel synthetic data set to compare the algorithms.

### 1.1. Main contribution of the paper

According to the 18th Article (Dough Rule) of the Land and Building Development Law (Web1, 2020; Web2, 2020), reallocation is carried out by giving the cadastral parcels from the zoning parcels that are hit in the zoning plan. In some cases, cadastral parcels can hit public spaces such as parks, green spaces, parking lots, health centers, educational establishments, municipal service areas and roads in zoning plans. In such a case, cadastral parcels that overlap public space are distributed to the nearest zoning parcels which are vacated by the development readjustment share (DRS) according to the 18th Article (Web2, 2020). As the distribution and subdivision processes have been already done manually, engineers and technicians who perform these operations face technical difficulties. In the manual distribution process, since the technician can act in a subjective manner, he or she can make unequal distribution by giving new zoning parcels near the old cadastral parcels to some parcel land owners and giving to the others from away. This will lead to the deterioration of social peace. Another problem is also that how and where to place the parcel will be in the zoning parcel plan. As an example; for the distribution of parcels that hit the public space, any parcel that is vacated from the DRS, near the relevant parcel is not always available. In such cases, the number of parcels which are divided into share will increase because the parcels to be distributed will be allocated to the parcels which are smaller than themselves. The shared parcel in the distribution is an undesirable situation. Because, the shareholders are not allowed to make any plans about that parcel. As far as the zoning parcels move away from the cadastral parcels, cadastral owners do not accept this situation. Therefore, they object to the existing zoning practice and cause the processes to delay for 3–12 months. This situation hinders the projects of investors who are going to construct in the application area. As a result, investments such as housing, industry and tourism and construction of public and commercial areas are delayed due to such problems in zoning applications.

This study has five main contributions to the literature:

**1. A new objective function for urban LR processes**: LR processes are difficult problems that are need to be implemented by considering different criteria simultaneously. These processes are carried out according to the 18th Article (Dough Rule) of the Land and Building Development Law in Turkey (Web1, 2020; Web2, 2020). Since these problems are currently handled by a technician or engineer, this situation reveals some different problems. The foremost of these problems is the uncertainty about whether the solutions obtained manually are objective or not. In order to overcome this problem, a new objective function is proposed, which tackles the general criteria in LR processes. With this objective function, solutions are evaluated objectively.

**2. Adaptation of swarm intelligence algorithms to the problem:** There are very limited technical studies for LR processes. These studies, on the other hand, concentrate more on the mathematical models. As far as our knowledge, this is the first study of urban LR processes adapted to a real-project site. The studies of Koc and Babaoglu can be said to be the closest study to this paper (Koc & Babaoglu, 2021). However, a synthetic data was used in the evaluations. In addition, in this study, three different *meta*-heuristic algorithms including Artificial bee colony (ABC), Particle Swarm Optimization (PSO) and Differential Evolution (DE) are used for solving the LR problem in a fast and effective way. These algorithms are tested comparatively under equal conditions. According to the experimental results, it can be seen that PSO is the most successful method and can be recommended for LR processes in real project sites.

**3. A development of an intelligent parcel-based crossover operator:** The LR problem is a discrete optimization problem. In such problems, crossover and mutation operators are required to solve the problem. However, the available crossover operators in the literature cannot be used directly because the problem has its own specific criteria. To deal with the LR problem, two different crossover operators are proposed in this study. The first crossover operator is the classical parcel-based crossover operator (CPC) which works completely randomly. The second operator is the intelligent parcel-based crossover operator (IPC) which runs with a greedy approach instead of using the random condition. According to IPC, the current situation and the post-crossover situation are compared, and if there is an improvement in the solution after the crossover, the crossover operation is performed. Otherwise, the crossover operation is canceled. Both of these methods are developed specific to the problem. However, considering that IPC produces better results than CPC, it is thought that the IPC technique can be used especially for real project sites. As a result, this operator can provide also a major contribution to the literature.

**4. Implementation of a new Expert system**: Zoning blocks on a real project site have different shapes and sizes. In the reallocation process, each owner can be allocated to a different block in each iteration and it is not possible that the total area of any block and the sum of the areas of all the owners to be allocated to that block are exactly the same in any iteration. This situation causes certain overflow or gap areas in the solutions obtained from the algorithms. Therefore, the solution obtained cannot be used directly. The overflow and gap areas on the blocks in the solutions obtained must be completely eliminated so that the solutions can be used in the real world. For this reason, a problem-specific expert system (LRES) is designed in this study. Thanks to this LRES, the overflow and gap areas are ranked in a certain order and the distribution of the owners corresponding to those areas is carried out again. Hence, all blocks have 100% occupancy rate and the results obtained are completely made applicable in the real life. Consequently, LRES can provide an important contribution to the literature, as well.

**5. Fully automated LR processes:** LR processes are currently carried out manually by technicians or experts. This situation has different disadvantages such as the prolongation of the project, inability to produce the expected output and difficulties in objective evaluation of the candidate solutions. Because of that, in this study, a system that can perform LR processes fully automatically is proposed. Firstly, the objective function is determined, then an artificial intelligence optimization algorithm is run considering this objective function. Finally, with the support of the expert system, the results are made directly applicable in the real world. As a result, the LR processes are carried out fully automatically, and the results/outputs are more objectively produced in a much shorter time compared to the manual method.

The rest of the paper is organized as follows: In Section 2, subdivision and reallocation problems in urban areas together with PSO, ABC and DE algorithms are presented. In Section 3, implementation of the *meta*-heuristic algorithms on solving subdivision and distribution problems are presented. The implementation of expert system is also described in Section 4. Experimental results and Discussion, and the conclusion of the paper are presented in Section 5, 6, respectively.

## 2. Material and method

### *2.1. Subdivision and reallocation problems in urban areas*

Subdivision and reallocation problems in urban areas are defined as the most appropriate settlement of all the owners in the cadastral parcels into the zoning blocks, taking into account the location of the cadastral parcel. The solution of these problems is realized according to the distribution and subdivision criteria as follow in Article 18th (Web2, 2020):

Owners in each cadastral parcel should be allocated to the nearest zoning parcel whenever possible. Besides, if the cadastral and the zoning parcel overlap, the owners must be allocated to that zoning parcel.

If the technical condition of the application area is suitable, the owners should be placed on the zoning parcel detachedly. In other words, the owner is placed on the parcel alone. If it is not suitable, the owners should be placed on the zoning parcel jointly. In other words, the owner is allocated to the parcel together with other owners.

The solution of the problem should provide all the zoning subdivision requirements such as all the parcels have the facade toward the road and that parcels must ensure the minimum facade lengths.

**Cadastral Blocks**: The current blocks in the project area represent cadastral block.

**Cadastral Parcels**: The current parcels in each cadastral block is named as cadastral parcel.

**Zoning Blocks**: The project area is divided into subareas by surrounding roads. Each subarea represents a zoning block.

**Owner List**: Each land owner in each parcel in the cadastral area is named as the owner. There can be one or more owners in a parcel. This list with all of the owners refers to the owner list. In the owner list, the total area of the owner is also given.

**Distribution List**: One owner can be located in more than one cadastral parcel. For example, if the owner named *A* has a total area of 500 $m^2$, for this owner, an area of 200 $m^2$ in the 3th parcel and 300 $m^2$ in the 15th parcel may be allocated. Therefore, one owner can have more than one distribution. The distribution list contains data such as the cadastral block, parcel, shared area, and shared ratio of the parcel for each owner.

### *2.2. Methods:*

#### *2.2.1. Artificial bee colony (ABC)*

Artificial Bee Colony algorithm (ABC) was proposed by Karaboga for real parameter optimization in 2005. ABC algorithm is an optimization algorithm inspired by the behavior of bee colonies in nature (Karaboga, 2005; Karaboga & Basturk, 2007; Karaboga, Gorkemli, Ozturk, & Karaboga, 2014; Marinakis, Marinaki, & Matsatsinis, 2009). The inspiration involves sharing knowledge within a colony of bees collecting honey in natural life. There are mainly three types of bees in the bee hive: employed bees, onlooker bees and scout bees. While the exploitation of the nectar source is carried out by employed bees and onlooker bees, the exploration of the search area is carried out by scout bees. In ABC optimization algorithm, the number of nectar sources is equal to the number of bees used. In addition, the number of employed bees is equal to the number of onlooker bees (Bansal, et al., 2011). ABC algorithm has

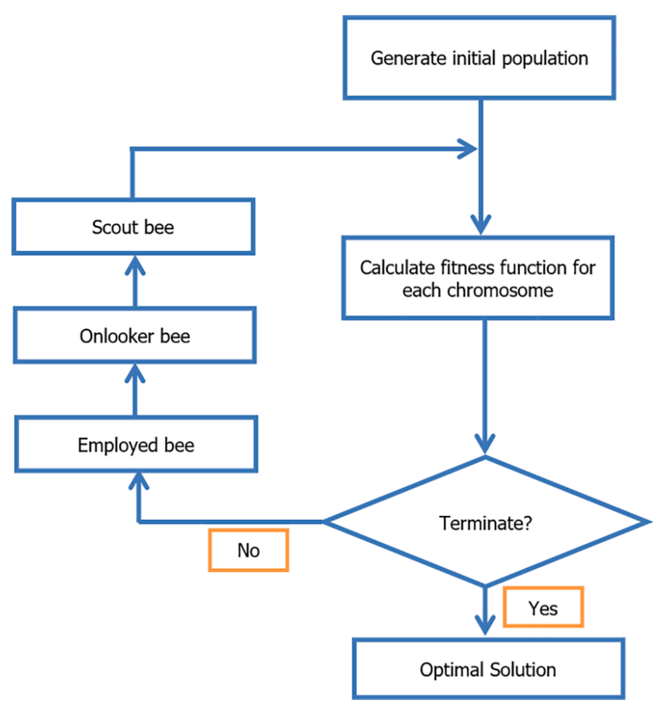


**Fig. 1.** Flow chart of ABC.

four stages; initialization, employed bee, onlooker bee and scout bee, and the flow chart of the algorithm is given in Fig. 1.

### 2.2.2. Particle swarm optimization (PSO)

Particle Swarm Optimization (PSO) algorithm which is based on the social behavior of organism groups such as herd, school and community has been proposed by Eberhart and Kennedy (Kennedy & Eberhart, 1995; Sammut & Webb, 2011). The position and velocity of each agent is represented by $P_j$ and by $V_j$ for $j^{th}$ dimension in the search space, respectively. Changing the position of the agents is performed by using position and velocity vectors. The PSO optimizes a specific objective function. Each agent knows his best value ever called '*Pbest*', which contains vectors about position and velocity. This information is an analogy to the personal experiences of each agent. In addition, each particle knows the best value until now, named "*Gbest*", among the *Pbest* values. Each agent tries to change its own position by considering their current positions ($P_x$, $P_y$), current velocities ($V_x$, $V_y$), individual intelligence called '*Pbest*' and group intelligence called '*Gbest*'.

Eq. (1) and (2) are used to calculate the position and velocity of particles in the PSO algorithm.

$$V_i^{t+1} = w^*V_i^t + c_1xrand_1x(Pbest_i - P_i^t) + c_2xrand_2x(Gbest - P_i^t) \tag{1}$$

$$P_i^{t+1} = P_i^t + V_i^{t+1} \tag{2}$$

In Eq. (1) and (2), $V_i^{t+1} and V_i^t$ represents velocity of $i^{th}$ particle at $(t + 1)^{th}$ *and* $t^{th}$ iteration, respectively. $c_1$ and $c_2$are constant variables which commonly take values in the range [0, 2]. $rand_1$and $rand_2$ express the numbers selected randomly in the range [0, 1]. $Pbest_i$ and *Gbest*represent the best values of $i^{th}$ particle and the best values among the swarm, respectively. Furthermore, $P_i^t$ represents position of $i^{th}$ particle at $t^{th}$ iteration. The velocity of each agent is changed according to Eq. (1), then the position of the agent is updated using Eq. (2). $w$ is a parameter that significantly affects the velocity updating and convergence of the particle, and many different techniques have been proposed in the literature for modification and usage of this parameter (Bansal, et al., 2011).

Besides, the flow diagram of the PSO algorithm is given in Fig. 2.

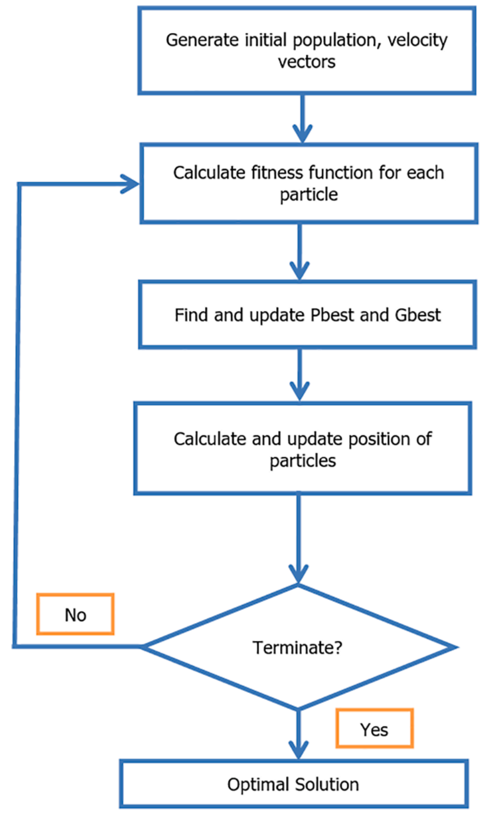


**Fig. 2.** Flow chart of PSO.

### 2.2.3. Differential Evolution (DE)

Differential Evolution (DE) algorithm is a population-based optimization technique proposed by Price and Storn in 1995, which is similar to genetic algorithm (GA) in terms of its operations and operators. The DE algorithm can produce effective results especially in continuous problems (Mayer, Kinghorn, & Archer, 2005; Storn & Price, 1997). Being able to make searches in many points in the search space at the same time, DE searches for better quality results in obtaining the solution of the problem with the help of its operators throughout the generations. Despite being similar to binary GA, unlike the classic binary GA, variables are represented with real values in DE algorithm (Hrstka & Kučerová, 2004). Although the crossover, mutation and selection operators in GA are used in the DE method, the way and order of operators differ from each other. In DE, chromosomes are handled one by one, and a new individual is generating by using three other randomly selected chromosomes. Mutation and crossover operators are used to perform these operations (Ali, Pant, & Abraham, 2009; Keskintürk, 2006). Flow chart of DE is given in Fig. 3.

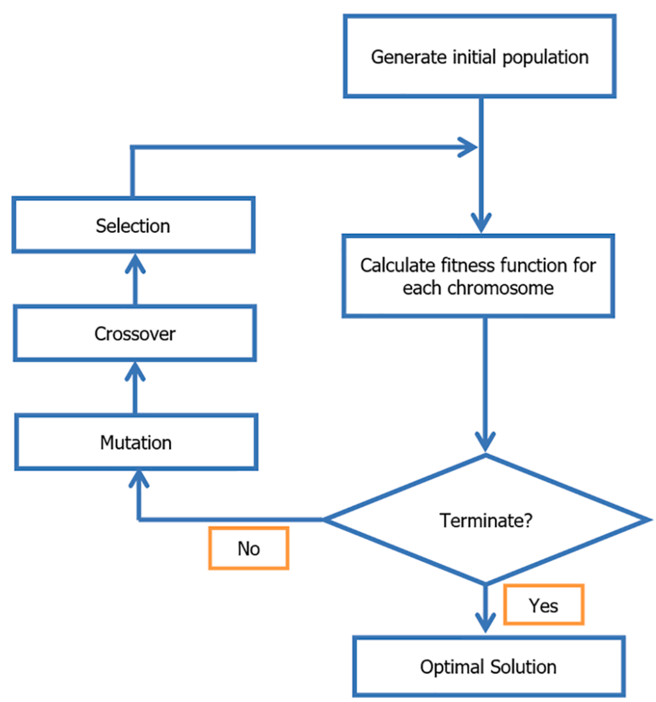


**Fig. 3.** Flow chart of DE.

## 3. Implementation of metaheuristic algorithms to solve subdivision and reallocation problems

The LR problems are very time-consuming, hard and complex real-world problems because they contain many different criteria to be handled. Because of this, utilizing classical methods is not sufficient to generate ideal the LR plans. In this study, the ABC, PSO and DE are implemented to evaluate simultaneously all of the goals and to reduce the computation time for the LR problem.

### 3.1. The proposed objective function

In order to solve the LR problem, the objectives required in solving the problem must be determined correctly. Thanks to the obtained objective function, the algorithms will calculate the solutions without the need for a specialist and will be able to decide which one is good or bad. Thus, better solutions can be used in the next generations.

There are two main objectives in the LR problem;

To minimize the number of joint shares of the owners who can technically form a detached parcel

To realize the allocation of the owner to the nearest zoning parcel to his/her cadastral parcel, in other words, to minimize the distance between the current cadastral parcel where the owner has already located and the zoning parcel determined by the algorithm for this owner.

Minimizing these two goals at the same time means approaching the ideal solution for the problem. The new parcels to be created on any block consist of different numbers of owners. However, on any real application site, the total area of these owners will not exactly equal to that block area. In other words, either gap or overflow will occur on the block in terms of area. Therefore, minimizing this gap or overflow area expected to occur on the block is determined as another goal for the algorithm. As a result, the proposed objective function will be run to minimize all the three aims simultaneously. The problem inputs and variables used in the proposed model are given below:

a. Problem inputs

nO: Number of owners
*nCB*: Number of cadastral blocks
*nZB*: Number of zoning blocks

$nPCB_i$: Number of parcels in $i^{th}$ cadastral block ($\forall\, i \in nCB$)

$nPZB_j$: Number of parcels in $j^{th}$ zoning blocks ($\forall$ j $\in nZB$)
*nTP*: Total number of parcels

$PA_{ij}$: area of $i^{th}$ parcel of $j^{th}$ zoning block ($\forall\, i \in nPZB_j$ , $\forall\, j \in nZB$)

$BA_i$: area of $j^{th}$ zoning block ($\forall$ j $\in nZB$)
DL: the longest diagonal length of the project

b. Decision variables

$D_{ijklm}$: For $m^{th}$ owner, the Euclidean distance between $k^{th}$ parcel of $i^{th}$ cadastral block to $l^{th}$ parcel of $j^{th}$ zoning block ($\forall$i $\in nCB$, $\forall$j $\in nZB$ , $\forall$k $\in nPCB_{\text{Ä°}}$, $\forall$l $\in nPZB_j$ ,$\forall$m $\in$ M)

$nJZP_m$: Number of joint zoning parcels although it is suitable to produce a detached parcel belonging to $m^{th}$ owner ($\forall$m $\in$ M)

$nZPS_m$: The total number of zoning parcel of $m^{th}$ owner ($\forall$ m $\forall$M)

$GR_i$: Overflow/gap ratio of $i^{th}$ zoning block

c. Constraints

At least one facade of each parcel must have a road front.

The parcels must meet the minimum facade length and edge length criteria according to the given plan.

Edge length of the corner parcel must be at least 14 m, and edge length of the inner parcel must be at least 12 m. These constraints are presented in detail in their studies of (Koc, et al., 2020).

d. Mathematical model of the proposed objective function

In this study, a novel mathematical model is proposed for LR problems. The mathematical model is formed considering the problem, its inputs, decision variables and constraints as given below.

$$TJ = \sum_{m=1}^{M} nJZP_m \tag{3}$$

$$D = \sum_{m=1}^{M} \sum_{i=1}^{nJZP_m} D_{m,i} = D_{ijklm} \tag{4}$$

$$GR_i = Abs\left(1 - \frac{\sum_{j=1}^{nPZB_i} PA_{ij}}{BA_i}\right. \tag{5}$$

$$GR = \sum_{i=1}^{nZB} GR_i \tag{6}$$

$$Z = \min\left(k1 \frac{100*TJ}{nO} + k2*\frac{D}{DL} + k3*\frac{100*GR}{nZB}\right. \tag{7}$$

$$0 < k1, k2, k3 < 1 \tag{8}$$

In Eq. (3), *TJ* stands for the total number of joint shares. While calculating the value of *TJ*, only the owners who have been allocated to the parcel jointly although they would be able to form a detached parcel are included to this value. For example, let's assume that an area of 400 $m^2$ is enough to generate a detached parcel. Although the land owner's area is 500 $m^2$, if this owner is distributed to a joint parcel, the *TJ* value is increased by 1. If the owner's area is below 400 $m^2$, any kind of the allocation of the owner has no effect on *TJ* value. Then, this value is divided by *nO* value in the objective function. Thus, a value between 0 and 1 is obtained. Finally, a value between 0 and 100 are produced by multiplying this value by 100 in Eq. (7).

*In Eq. (4), D* denotes the sum of distance values between the cadastral parcel and zoning parcels of all the owners in the project. *DL* value is a project-specific input parameter, and this value is the distance of the two most distant points in the project. Thus, even if the project site and project scale changes, the objective function will work effectively in the project. This value obtained by making *D/DL* is reflected to the objective function in Eq. (7). Since the values obtained by *D/DL* operation are between 10 and 100, any multiplier is not necessary.

*GR* represents the sum of gap or overflow rates on the zoning blocks in *In Eq. (6)*. For example, if any block area is 3000 $m^2$ and the sum of the areas of the parcels in the blocks is 3500 $m^2$, then *Abs*(1 - (3500/3000)) is done when calculating $GR_i$ *In Eq. (5)*. $GR_i$ value is normally calculated as −0.2 without *Abs* function which is absolute value function. However, positive value is found thanks to *Abs* function. Therefore, overflow or gap on the block is considered equal for the objective function. This situation always ensure the objective function to run correctly. In this calculation, since only the values between 0 and 1 will be obtained, it is also converted to a number between 0 and 100 by multiplying this value by 100 in Eq. (7). Thus, these three penalty values in the goal function will be able to produce effective results without dominating each other. Finally, these values are multiplied by the scaling parameter predefined by user and cost (*Z*) is calculated according to Eq. (7). The *k1*, *k2* and *k3* values are scaling values in Eq. (8). In this study, these values are taken equally as 1/3.

### 3.2. Generating initial population

Algorithms need a population to run throughout generations. The pseudo code of generating initial population is given in Fig. 4. Algorithm

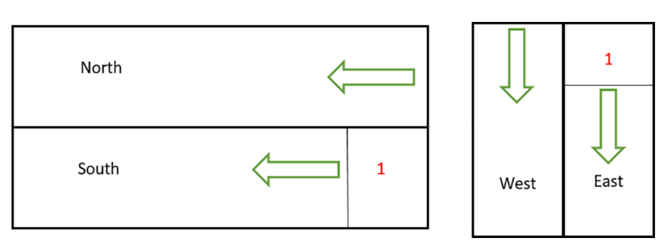


**Fig. 4.** An example of creating a parcel in the blocks according to the direction.

1.

**Algorithm 1.** Pseudo code of generating initial population

***Procedure*** *Generate Initial Population()****Begindo***

*Choose randomly a zoning block (Block$_i$).*

***do***
*Select a random land owner (Owner$_m$).*
***while*** *the parcel area to be generated does not meet the minimum area condition.*

*Allocate this owner (Owner$_m$) starting from the south east of the selected block (Block$_i$).*

***if*** *Block$_i$ is a horizontal block* ***then***
*Start creating parcel from South direction: Look at*

***else If*** *Block$_i$ is a vertical block,* ***then***
*Start creating parcel from East direction: Look at*

***end else if***
***end if***

*Create zoning parcel using binary search method.*

*if* *There is more than one owner in the parcel to be created now* ***then***
*Create a joint parcel and allocate these owners in this parcel*

***else if*** *there is only a single owner* ***then***
*Create a detached parcel and allocate this owner in this parcel*

***end else if***
***end if***

***if*** *the first part of Block$_i$ is completed* ***then***
*Move to the second part of Block$_i$* ***end Ifwhile*** *All owners are not allocated in zoning parcels*
***Return*** *blockList*
***End***

### 3.3. Evaluating the initial population

The cost value of each individual in the population is evaluated according to the objective function. For this process, the blocks belonging to each individual and the parcels in these blocks and the owner or owners in each parcel are sent to the objective function in order to evaluate solution quality of each individual.

### 3.4. Crossover Operator: Parcel-based crossover

Crossover is a process to recombine two different parents. New offspring occurs at the end of the crossover. In this study, a new spatial data-based crossover is recommended specific to the nature of the problem, unlike crossover methods in the literature. The crossover process is carried out by crossing the parcels with the same indexes in the same block in the other individual/parent corresponding to the parcels in the blocks. An example of the proposed crossover method is shown in Fig. 5. In the example, two different individuals are randomly determined. Each individual consists of two blocks with horizontal and vertical geometric structures. 18 owners with the different areas are determined. By assigning these owner or owners randomly to the parcels, the parcel sequence shown in Fig. 5 are formed by the proposed crossover operator. In addition, two different crossover techniques are presented in this study and these operators are in detail explained in Section 3.4.1 and 3.4.2.

#### 3.4.1. Classical Parcel-Based crossover (CPC)

The classical crossover (CPC) method simply works randomly. According to the example given in Fig. 5, the 4th parcels of the 1st block are determined to crossover. As can be seen, there are owners named *N* and *O* in the parcel determined in *individual2/parent2*. Since only owner *O* of *N* and *O* owners provides a random situation, crossover process is applied for this owner. Then the parcel of *O* owner in *individual1/parent1* is found, and this owner is removed from this parcel. After crossover, there are owners named *G*, *J* and *O* in the 4th parcel of the 1st individual.

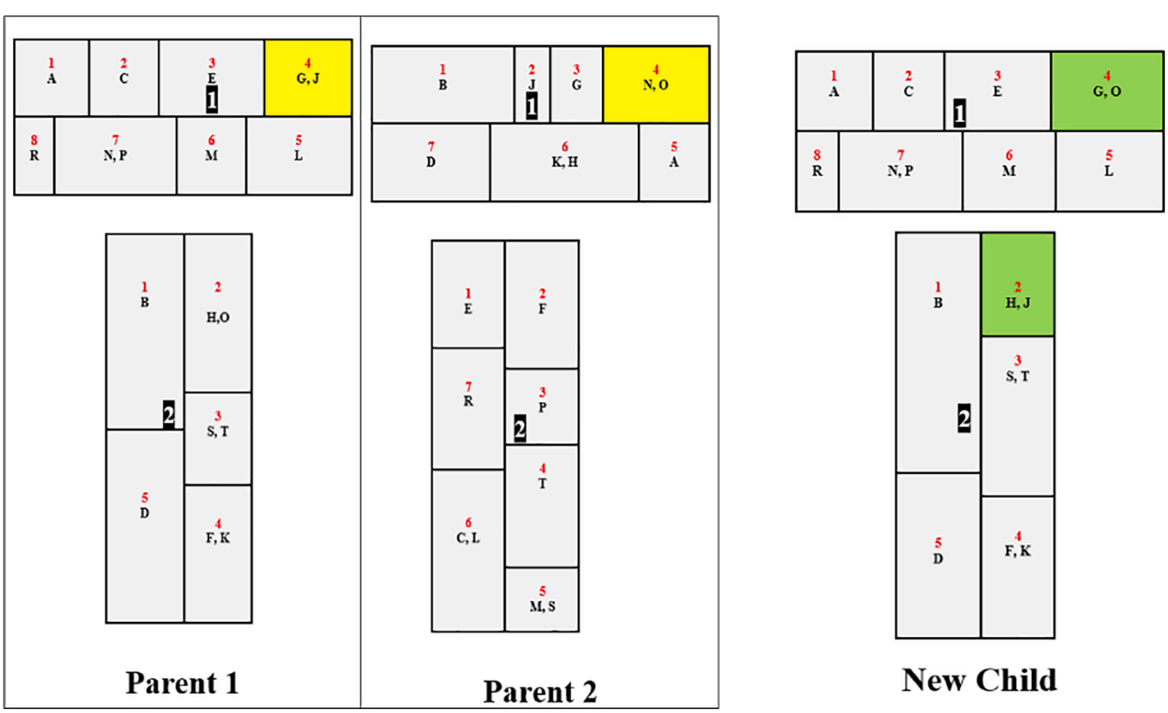


**Fig. 5.** An example of parcel-based crossover (General Visualization).

At this stage, the owners who provide the random situation are sent to the 2nd parcel of the 2nd block. According to this example, since only owner *J* meets this condition, this owner is sent to 2nd parcel by removing him/her from the current parcel. As a result, the areas and list of owners of these two parcels are updated according to the change (insertion/deletion) in owner allocations. This process, which is shown in Fig. 5, is given for only a single parcel. The classical crossover operator is given in detail in Algorithm 2.

**Algorithm 2.** Pseudo Code of *Classical Parcel-Based Crossover*

```
Procedure Classical Parcel-Based Crossover()
Begin

CRrate = Determine crossover rate ();
NP = the number of parcels in blocks of individual1();
NCP = NP*CRrate ; // the number of parcels to crossover according to the crossover rate
parcelList = Select parcels with the same block of the same index from two different
individuals randomly();
parcelList1 = Get parcel List of Individual1(); // parcels of individual1 in parcelList
parcelList2 = Get parcel List of Individual2();// parcels of individual2 in parcelList

for i = 1 to NCP do
  parcel1 = parcelList1[i]
  parcel2 = parcelList2[i]
  owners1 = Get Owners of parcel1();
  owners2 = Get Owners of parcel2();

for each owner2 in owners2 do

if rand < rand then

  owners1.Add(owner2) //Add owner to distribution list and Allocate owner in parcel
  parcel1.area = parcel1.area + owner2.area // Update area of parcel
  otherParcel = Find parcel of owner2 in blocks of individual1();
  otherOwners = Get Owners of otherParcel(); // All of the Owners of otherParcel;
  otherOwners.Remove(owner2) // Remove owner from distribution list and Delete owner
from otherParcel
  otherParcel.area = otherParcel.area – owner2.area // Update area of parcel

for each owner1 in owners1 do

  if rand < rand then
    otherOwners.Add(owner1) // Add owner to distribution list and Allocate owner in
  parcel
    otherParcel.area = otherParcel.area + owner1.area // Update area of parcel
    owners1.Remove(owner1) // Remove owner from distribution list and Delete owner
  from parcel1
    parcel1.area = parcel1.area- owner1.area // Update area of parcel
      end if
    end for
  end if
  end for
end for

Return Individual1

End
```

### 3.4.2. Intelligent Parcel-Based crossover (IPC)

Unlike the classical crossover operator (CPC), the intelligent crossover operator (IPC) works to serve two main targets in the objective function. The first one is to allocate the owner in the closest zoning parcel to his/her position in the cadaster, and the other one is to minimize this distance where the current state of the parcel and the new destination parcel are taken into consideration. If the new position is closer to that of the cadastral, crossover is performed. The second goal is reducing the number of shares. Hereby, if there is a suitable subset for the parcel or parcels exposed to crossover providing technical conditions, then the existing parcel is divided and a new parcel is added to reduce the number of shares.

According to the example which is given to demonstrate the IPC in Fig. 5, the 4th parcels of the 1st block are determined to crossover. As can be seen, there are owners named *N* and *O* in the parcel determined in *parent2*. Since only owner *O* of *N* and *O* owners provides distance condition, crossover process is applied for this owner. Then the parcel of *O* owner in *parent1* is found and this owner is removed from this parcel. After crossover, there are owners named *G*, *J* and *O* in the 4th parcel of *parent1*. At this stage, the owners who provide the distance situation are sent to the 2nd parcel of the 2nd block. Since only owner *J* meets this condition, this owner is sent to 2nd parcel by removed from his/her current parcel. As a result, the areas and list of owners of these two parcels are updated according to the change (insertion/deletion) in owner allocations.

After this stage, new updated parcels focus on reducing the number of shares. If the parcel can be split technically (if it has enough area for formation of a parcel of minimum area), the area and distribution list of the existing parcel are updated according to the first subset of the owners. Second owners are also used to create new parcel. The new parcel is created adjacent to the existing parcel. Then, new parcel's area is calculated according to the distribution list allocated for this parcel. This operation is presented in Fig. 6. This process which is shown in Fig. 5 is described for only a single parcel. The detailed pseudo code of the IPC operator is given in Algorithm 3.

***Algorithm 3.*** Pseudo Code of *Intelligent Parcel-Based Crossover*

```
Procedure Intelligent Parcel-Based Crossover()
Begin
cadastralList = Get Cadastral Data();
CRrate = Determine crossover rate ();
NP = the number of parcels in blocks of individual1();
NCP = NP*CRrate ; // the number of parcels to crossover according to the crossover rate
parcelList = Select parcels with the same block of the same index from two different
individuals randomly();
parcelList1 = Get parcel List of Individual1(); // parcels of individual1 in parcelList
parcelList2 = Get parcel List of Individual2();// parcels of individual2 in parcelList

for i = 1 to NCP do
```

(continued on next page)

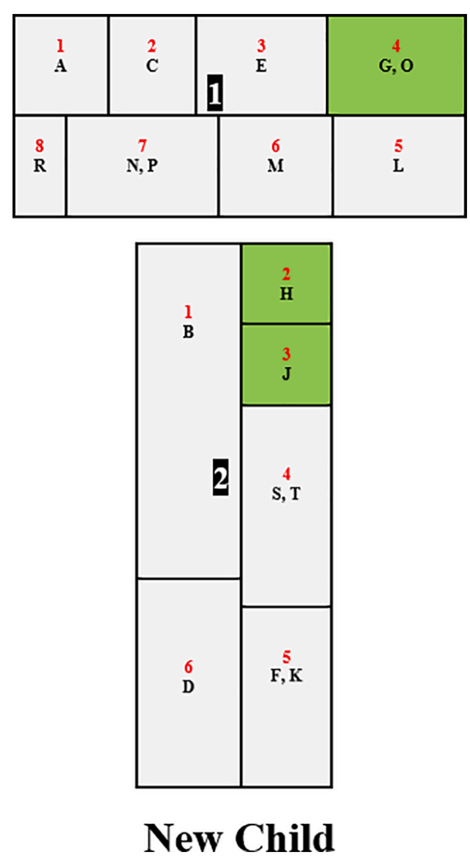


**Fig. 6.** An example of new parcel formation of IPC.

(*continued*)

**Algorithm 3.** Pseudo Code of *Intelligent Parcel-Based Crossover*

```
parcel1 = parcelList1[i]
parcel2 = parcelList2[i]
owners1 = Get Owners of parcel1();
owners2 = Get Owners of parcel2();

for each owner2 in owners2 do

  positionOfOwner2 = Get position of owner2 in cadastralList();
  currentDistance = Calculate distance between positionOfOwner2 and parcel2();
  newDistance = Calculate distance between positionOfOwner2 and parcel1();

    if newDistance < currentDistance then // if the distance is decreasing compared to its current position

    owners1.Add(owner2) //Add owner to distribution list and Allocate owner in parcel
    parcel1.area = parcel1.area + owner2.area // Update area of parcel
    otherParcel = Find parcel of owner2 in blocks of individual1();
    otherOwners = Get Owners of otherParcel(); // All of the Owners of otherParcel;
    otherOwners.Remove(owner2) // Remove owner from distribution list and Delete owner from otherParcel
    otherParcel.area = otherParcel.area – owner2.area // Update area of parcel

for each owner1 in owners1 do

    positionOfOwner1 = Get position of owner1 in cadastralList();
    currentDistance = Calculate Distance between positionOfOwner1and parcel1();
    newDistance = Calculate Distance between positionOfOwner1and otherParcel();

    if newDistance < currentDistance then // if the distance is decreasing compared to its current position
    otherOwners.Add(owner1) // Add owner to distribution list and Allocate owner in parcel
    otherOwners.area = otherOwners.area + owner1.area // Update area of parcel
    owners1.Remove(owner1) // Remove owner from distribution list and Delete owner from parcel1
    parcel1.area = parcel1.area- owner1.area // Update area of parcel
end if
end for
end if

  Create New Parcel(parcel1); // Create new parcel from parcel1
  Create New Parcel(otherParcel); // Create new parcel from otherParcel
end for
end for
Return Individual1
End
Procedure Create New Parcel(myParcel)
Begin
CombList = Get all combinations of owners of myParcel for producing two separate parcels ();

for each comb_i in CombList
  owners1 = Get First Owners of comb_i();
  owners2 = Get Second Owners of comb_i();

if both of owners1 and owners2 is met to produce two separate parcels then
  myParcel.area = sum of the area of owners1; // Update myParcel
  myParcel.DistributionList = owners1;
      newParcel = Create parcel(); //Create New Parcel
  newParcel.area = sum of the area of owners2;
  newParcel. DistributionList = owners2;
  break;
end
End
```

### 3.5. Mutation Operator: Minimizing gap/overflow of blocks

Mutation process is applied to form some changes in available individual after crossover operator. The mutation operator is used to prevent early convergence and to prevent stuck to the local optimal by replacing certain genes of an individual by help of some possibilities (Aggarwal, Saxena, Back, & Emmerich, 2020). In this study, two different mutation operators are utilized in accordance with spatial data.

#### 3.5.1. Move parcels in the same block

In this mutation operator, the goal is to minimize overflow and gap areas in the blocks in terms of area. Firstly, the overflow/gap areas of both parts of the blocks are calculated. If there is a free area in one direction of the block and an overflowing area in the other direction and these areas are more than the minimum area, the following steps are applied.

1. Calculate the overflow/free areas of both parts of the blocks.
2. Determine part with overflowing areas (*P1*) and part with free areas (*P2*) of the block.
2. Sort the parcels in *P1* from small to large in terms of area.
3. Then transfer/move these parcels to *P2*.
4. Terminate this process (Step3) when overflowing areas of *P1* approach to zero and free areas of *P2* approach to zero. (If another parcel is transferred to *P2*, this time there will be a gap in *P1*, and an overflow in *P2*)

#### 3.5.2. Move parcels in the different blocks

In this mutation operator, the goal is to balance two different blocks in terms of area. The process steps are as follows.

1. Firstly, the overflow/free areas of both parts of the blocks are calculated.
2. Find the block directions with the highest overflow (*B1*) and the largest gap (*B2*) within all the blocks.
3. Sort the parcels in *B1* by small to large area.
4. Then transfer/move these parcels to *B2*.
5. Terminate this process (Step4) when overflow areas of *B1* approaches zero and free areas of *B2* approach to zero. (If another parcel is transferred to *B2*, this time there will be a gap in *B1*, and an overflow in *B2*).

### 3.6. Combining unsuitable parcels

At this stage, technically unsuitable parcels are combined to create suitable parcels. In the crossover step of the algorithm, there is no areal constraint and thus the protection of genetic diversity is aimed. The absence of such a restriction in the crossing step can lead to the formation of unsuitable plots. In order to solve this problem, this step is added to the algorithm, and unsuitable parcels are combined with neighboring parcel or parcels to obtain a new and larger parcel.

### 3.7. Updating subdivision plan

At this stage, it is necessary to determine the new location of the parcels of the blocks considering the updated distribution plan. For this process, Binary Search method, which gives very fast and effective results, is used (Koc, et al., 2020).

### 3.8. Evaluating new candidate solution

Using the updated distribution and subdivision plans, the new candidate solution is sent to the objective function and the new cost value is calculated.

### 3.9. Terminating the algorithm

If the algorithm reaches to the maximum number of iterations, it is terminated and the best solution obtained is reported.

In addition, in this study, the flowchart of the proposed method used to obtain new individuals is given in Fig. 7. As can be seen, all steps are applied in each process. However, only one of crossover operators is run in one process and crossover rate is defined by user. In the *crossover* process mentioned in PSO, ABC and DE algorithms, these process steps are implemented as shown in the Fig. 7.

Implementation of PSO, ABC and DE which are used for solving LR problems are given in Algorithm 4, 5 and 6, respectively. Firsty, *N*

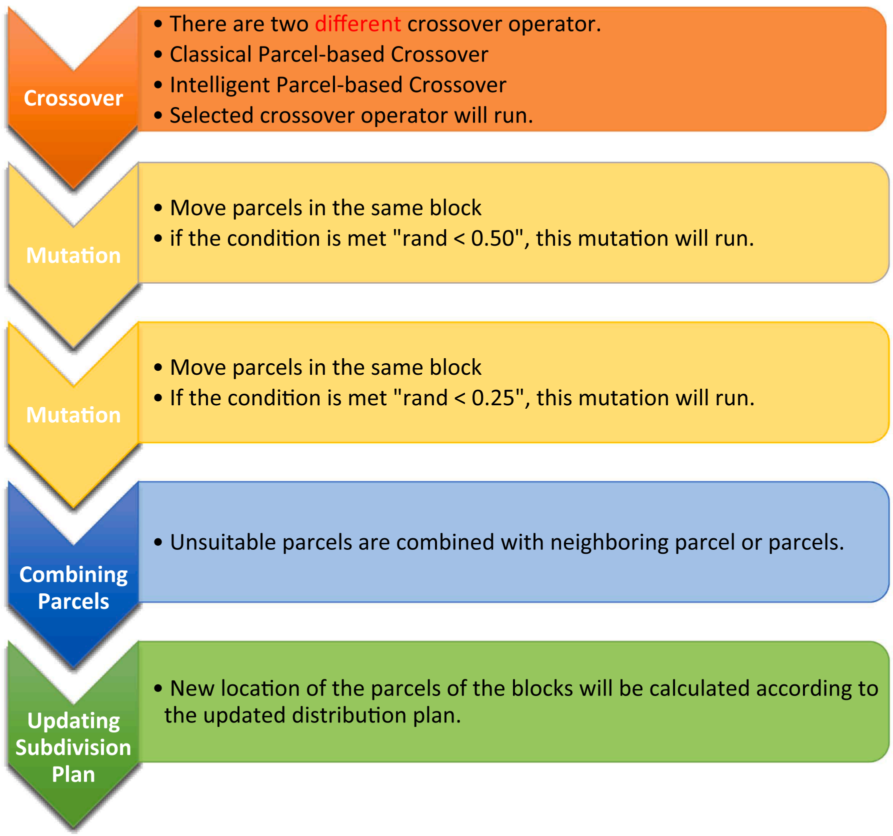


**Fig. 7.** The flowchart of the proposed method containing crossover and mutation operators.

number of individuals are produced for algorithms. Then, these individuals are subjected to the crossover process according to the predefined crossover rates, and then, new individuals are produced. After that, these individuals are transferred to the next generations according to the structure of the algorithms.

**Algorithm 4.** Pseudo code of implementation of PSO to LR problem

***Prodecure*** *Implementation of PSO to LR()*
***Begin***

*Generate randomly particle population();*
$N$ = *number of particle;*
*Evaluate cost of particle population();*
*Pbest = particle;*
*Find Gbest;*
$iter = 1;$

***while*** $iter < MaxIter$ ***do***
  ***for*** $i = 1$ to $N$ ***do***
  *Crossover particle$_i$ with Pbest$_i$ and create a new particle (p1).*
  *Crossover particle$_i$ with Gbest and create a new particle (p2).*
  *Crossover p1 with p2 and create a new particle (p3).*
  *Evaluate cost of p3.*

  ***if*** *new cost is better than that of the Pbest$_i$* ***then***
    *Update Pbest$_i$*
  ***end***
  ***if*** *new cost is better than that of the Gbest* ***then***
    *Update Gbest*
  ***end***
  $iter = iter + 1;$
***end***
***End***

**Algorithm 5.** Pseudo code of implementation of ABC to LR problem

(continued on next column)

(continued)

**Algorithm 5.** Pseudo code of implementation of ABC to LR problem

***Prodecure*** *Implementation of ABC to LR()*
***Begin***

*Generate randomly food sources();*
$N$ = *number of food source;*
*Evaluate cost of food sources();*
$iter = 1;$

***while*** $iter < MaxIter$ ***do***
***for*** $i = 1$ to $N$ ***do*** *//Employee bee*
  *Crossover the randomly selected food source with the food source$_i$ and create a new food source.*
  *Evaluate cost of new food source.*
  ***if*** *new cost is better than that of the food source$_i$* ***then***
    *Change new food source with the food source$_i$*
    $Limit = 0$
  ***else***
    $Limit = Limit + 1$
  ***end***
  ***end***
***for*** $i = 1$ to $N$ ***do*** *// Onlooker Bee*
  *Crossover the food source selected from the Tournament method with the food source$_i$.*
  *Produce a new food source.*
  *Calculate the cost value of the new food source.*

***if*** *new cost is better than that of the food source$_i$* ***then***
  *Change new food source with the food source$_i$*
  $Limit = 0$
***else***
    $Limit = Limit + 1$
  ***end***
***end***

  ***for*** $i = 1$ to $N$ ***do*** *// Scout Bee*
    ***if*** *limit value of the food source$_i$ is greater than Limit* ***then***
      *Find a new food source.*

(continued on next page)

(*continued*)

**Algorithm 5.** Pseudo code of implementation of ABC to LR problem

```
    end
  end
  iter = iter + 1;
end
End
```

**Algorithm 6.** Pseudo code of implementation of DE to LR problem

```
Prodecure Implementation of DE to LR()
Begin

Generate randomly chromosomes();
N = number of chromosome;
Evaluate cost of chromosomes();
iter = 1;

while iter < MaxIter do
  for i = 1 to N do
    [C1, C2, C3] = Choose randomly three different chromosomes that are different from
chromosome_i
  Crossover C1 with C2 and create a new particle (newC1).
  Crossover C3 with newC1 and create a new particle (newC2).
  Crossover chromosome_i with newC2 and create a new particle (newC3).
  Evaluate cost of newC3.

  if new cost is better than that of the chromosome_i then
    Update chromosome_i
end
iter = iter + 1;
  end
end
End
```

## 4. Implementation of an expert system (LRES)

In this study, metaheuristic algorithms are proposed for solving LR problems. As given in the mathematical model, there are three main objectives to be addressed in the LR problem: 1. minimizing the number of shareholders, 2. minimizing the distance between cadastral and zoning parcel for each owner and 3. minimizing overflow or gap areas in the blocks. The first two of these goals are the main goals that should be minimized. The third aim is added to the objective function so that the algorithm can work flexibly, by don't increasing the number of shares after each iteration. In the real world, in order to be able to directly use the distribution and subdivision maps/outputs obtained by the optimization algorithms, overflows and gap areas in the blocks must be completely removed. In this study, an expert system specific to LR problems (LRES) is proposed to realize the occupancy rate of all blocks in the project area as 100%. Algorithm 7 presents the pseudo code of the proposed Expert System which is employed to balance block areas. According to Algorithm 7, the blocks with the most overflow and gap areas are determined, and these two blocks are operated according to 4 different cases in LRES. In each case, the gap and overflow areas are reduced or completely removed. In this study, the sum of overflow and gap areas in all the blocks are given to be maximum 10 $m^2$ as the stopping criteria. Considering that the project area is about 5 ha, it is quite clear that this area is negligible.

**Algorithm 7.** Pseudo code of Expert System to balance block areas

```
Procedure Expert System()
Begin
do
Calculate the occupancy of each block(); (each direction is evaluated separately).
Sort the blocks by occupancy rate();
B1 = Get Block with the highest occupancy rate(); // (Highest Overflow area)
B2 = Get Block with the lowest occupancy rate(); //(Highest gap area)
a1 = overflow area of B1
a2 = gap area of B2
P = Get the last parcel in B1();
Pa = area of P

  if Pa < a1 then // Sample: Case A and B : If Pa (500 m^2) < a1 (1200 m^2)

if Pa < a2 then // Sample: Case A: If Pa (500 m^2) < a2 (1500 m^2) then a2 will be 1000
m^2 and a1 will be 700 m^2.
  Remove P completely from B1();
  Add completely P to B2();
else // Sample: Case B: If Pa (500 m^2) > a2 (300 m^2) then a2 will be 0 and a1 will updated
as 200 m^2.
  Update Subdivision and Distribution (a2);
  Occupancy rate of B2 is updated as 100%.
end

else // Sample: Case C and D : If Pa (500 m^2) > a1 (400 m^2)

if a1 < a2 then // Sample: Case C: If a1 (400 m^2) < a2 (1000 m^2) then a1 will be 0 and
a2 will updated as 600 m^2.
Update Subdivision and Distribution (a1);
Occupancy rate of B1 is updated as 100%.
else // Sample: Case D If a1 (400 m^2) > a2 (300 m^2) then a2 will be 0 and a1 will updated
as 100 m^2.
Update Subdivision and Distribution (a2);
Occupancy rate of B2 is updated as 100%.
end

Update a1 and a2();

while sum of overflow or gap areas in all the blocks > 10 m^2

Return Subdivision and Distribution;

End

Procedure Update Subdivision and Distribution (parcel area)
Begin

a = parcel area;
O1 = Get the owners of P according to the area of each owner in order from large area to
small.
O2 = [];

while O1[0] < Pa1 do
  Add O1[0] to list O2();
  a = a- area of O1[0];
  Remove O1[0] from O1(); // O1[0] is going to next owner in each step.
end

area of O1[0] = a; // Update area and take the land/area from the owner as much as the
remaining area.
Add O1[0]to O2();

if Pa1 > min Area then // min Area is minimum area required for formation of a parcel in
the block
  Produce new parcel in B2.
  LP = new parcel;
else
LP = last parcel in B2;
end

Add O2 to LP.
Update area of LP.
Update P's area and distribution list.

End
```

## 5. Experimental results and Discussion

In this study, ABC, DE and PSO algorithms which are most widely used *meta*-heuristic approaches in the literature were implemented to solve the LR problem. All experimental studies are run on a machine with Intel Core i7 2.80 GHz CPU, 16 GB RAM and Windows 10 64-bit operating system, and all code implementations are carried out by C# programming language on Visual Studio 2019 software development environment.

The proposed method is implemented over a real project site, hometown of Harmancık of Konya in Turkey–whose LR process had already been completed. The input of the application area are given in Table 1. According to these data, the public land use deduction rate is calculated as 12.43%, 6954.64 $m^2$. The distribution and subdivision processes are applied to an area of 48,957.23 $m^2$. In addition, the cadastral and zoning maps of the project site is given in Fig. 8. There are also 113 land owners in this application.

In order to compare the algorithms fairly, parameter values are chosen equal together. The maximum number of evaluation (MaxFEs) value is determined as 20.000 for each algorithm and all algorithms are run 20 times independently with random seeds. The values obtained are presented in Table 3

All the parameters of the algorithms used in experimental studies is given in Table 2. In ABC algorithm, *Limit* value is selected as 50 and selection operator is determined as *tournament* method which are the algorithm's kernel parameters. The other parameters in the algorithms are taken with equal values.

Fig. 9 shows the formal subdivision plan obtained manually by an expert. In this plan, there are a total of 67 urban parcels. While there are 113 owners in this project area, these owners are allocated to the parcels in the zoning plan as a total of 196 shareholders.

Table 3 presents comparative results of the metaheuristic algorithms utilized in this study in terms of performance. When the attained results are analyzed, it is seen that the PSO algorithm shows a very significant success compared to other algorithms. It is also clear that when IPS crossover technique is used in the PSO algorithm, the algorithm acquires much better results than the results obtained by using the CPC method. When the robustness of the algorithms is observed, it is seen that the PSO algorithm is more robust than the other algorithms. When the results of algorithms supported by expert system are examined, it is clearly seen that PSO method produces more effective results than ABC and DE algorithms. When IPC and CPC crossover operators are compared on the basis of algorithms, it is seen that each algorithm shows a more effective performance than CPC when IPC crossover technique is used in the algorithm. When *MinCost* values are analyzed, it is seen that performance of the algorithms increases 2–3 times especially in DE and PSO methods. When *AvgCost* values are examined, it is seen that the performance of all algorithms increases two times. As a result, among the algorithms, PSO with IPS and PSO supported by expert system with IPS performs the best performance with values of 3.929, 6.652, respectively.

In terms of time, the comparative results of the metaheuristic algorithms are presented in Table 4. When the general results are analyzed, it can be seen that the algorithms evaluate 20.000 different solutions within maximum 10–13 min. Also, when looking at the times in crossover operators, it can be seen that the IPC method produces results in a slightly longer period due to the additional decision making steps compared to the CPC. Consequently, considering that it takes an average of 3 days to complete it when this project is carried out manually by the expert, it is very important for the place of the study in the literature that the algorithms yield effective results in as short as 5–15 min.

Figs. 10, 11 and 12 show the convergence graph of best results obtained by ABC, DE and PSO algorithms, respectively. In these figures, *Gap (GR)* values represents both of the overflow and space areas in all the blocks in the project site as in Eq. (5) and (6). In addition, *Share (TJ)* and *Distance (D)* penalty values is calculated according to Eq. (3) and (4), respectively. Cost ($Z$) values is evaluated using Eq. (7). In this study, $k1$ and $k2$ and $k3$ values are taken as equal which of each is 1/3.

When Fig. 10 is examined, convergence in the CPC method which is named as *Classical crossover operator* stops at around 600th iterations, while in the IPC operator which is named as *Intelligent CR*, convergence continues until 700th iteration. After these iterations, no meaningful improvement has been seen. In addition, it can be seen that the IPC method produces much more successful results in terms of *Cost* which is the general output of the objective function. In the CPC, it can be seen that the *Gap* penalty values converges better results than *Distance* and *Share* aim values. There is especially not much improvement in *Distance* penalty values of CPC. As for IPC, it is observed that both *Share* and *Gap* penalty values have similarly effective convergence in IPC. Finally, in IPC, all the values in objective function is seen to exhibits a good convergence.

When Fig. 11 is analyzed for the DE algorithm, the convergence result in the CPC operator continued up to about 900th iteration, while in the IPC operator, it is clearly seen that the convergence continued throughout the iteration. In addition, in the DE method, as in ABC, the IPC method gives much better results in terms of *Cost*. When evaluated according to the objective values, it can be seen that *Gap* penalty values exhibit the best convergence in both crossover techniques while *Distance* penalty values show the most unsuccessful convergence. In the IPC, it can be seen that the *Share* values is acquired close results to the *Gap* penalty value. As a result, it can be seen that IPC method is more successful than the CPC in terms of three different objective values. In addition, it can be stated that if number of the iterations is increased, the DE can achieve better quality solutions when using the IPC.

When the PSO algorithm is examined according to Fig. 12 for two different crossover techniques, it is observed that the CPC and the IPC operator continued to converge until about 900th iteration. By using the CPC operator, it can be seen that the *Gap*, *Share* and *Distance* penalty values are sorted in terms of convergence success. Although there is also the similar situation in the IPC operator, the IPC operator is able to bring the *Share* objective value to the most optimal value. It is also clear that the values of *Gap* in PSO with the IPC converge more successfully than any other algorithms. When both techniques are compared, similar to ABC and DE, it can be seen that the IPC operator achieves very good results in terms of *Cost* results compared to the CPC. It is also clear that PSO with the IPC operator is the algorithm that shows the fastest convergence within the first 100 iterations. As a result, it can be said that PSO with IPC is the best successful among all of the algorithms.

The general analysis of different techniques is comparatively presented in Table 5. This table shows the best results obtained from 20 independent runs of each algorithm. In addition to the *Cost* values of each algorithm, the objective values included in the objective function are also given. Moreover, the number of parcel and shareholders are also

**Table 1**
Cadastral and zoning data of the application area.

| Cadastral data | | | | | | Zoning data |
|---|---|---|---|---|---|---|
| Block Number | Block Name | Number of Parcel | Block Area ($m^2$) | Block Number | Block Name | Block Area ($m^2$) |
| 1 | 26,898 | 3 | 1,886.20 | 1 | 37,369 | 15,102.49 |
| 2 | 26,899 | 4 | 2,138.58 | 2 | 37,370 | 11,693.97 |
| 3 | 26,900 | 6 | 2,717.98 | 3 | 37,371 | 8,435.20 |
| 4 | 27,093 | 11 | 5,431.98 | 4 | 37,372 | 2,970.58 |
| 5 | 27,862 | 1 | 3,684.11 | 5 | 37,373 | 4,869.46 |
| 6 | 27,863 | 15 | 40,053.02 | 6 | 37,374 | 5,885.59 |
| | **Total:** | **40** | **55,911.87** | | **Total:** | **48,957.23** |

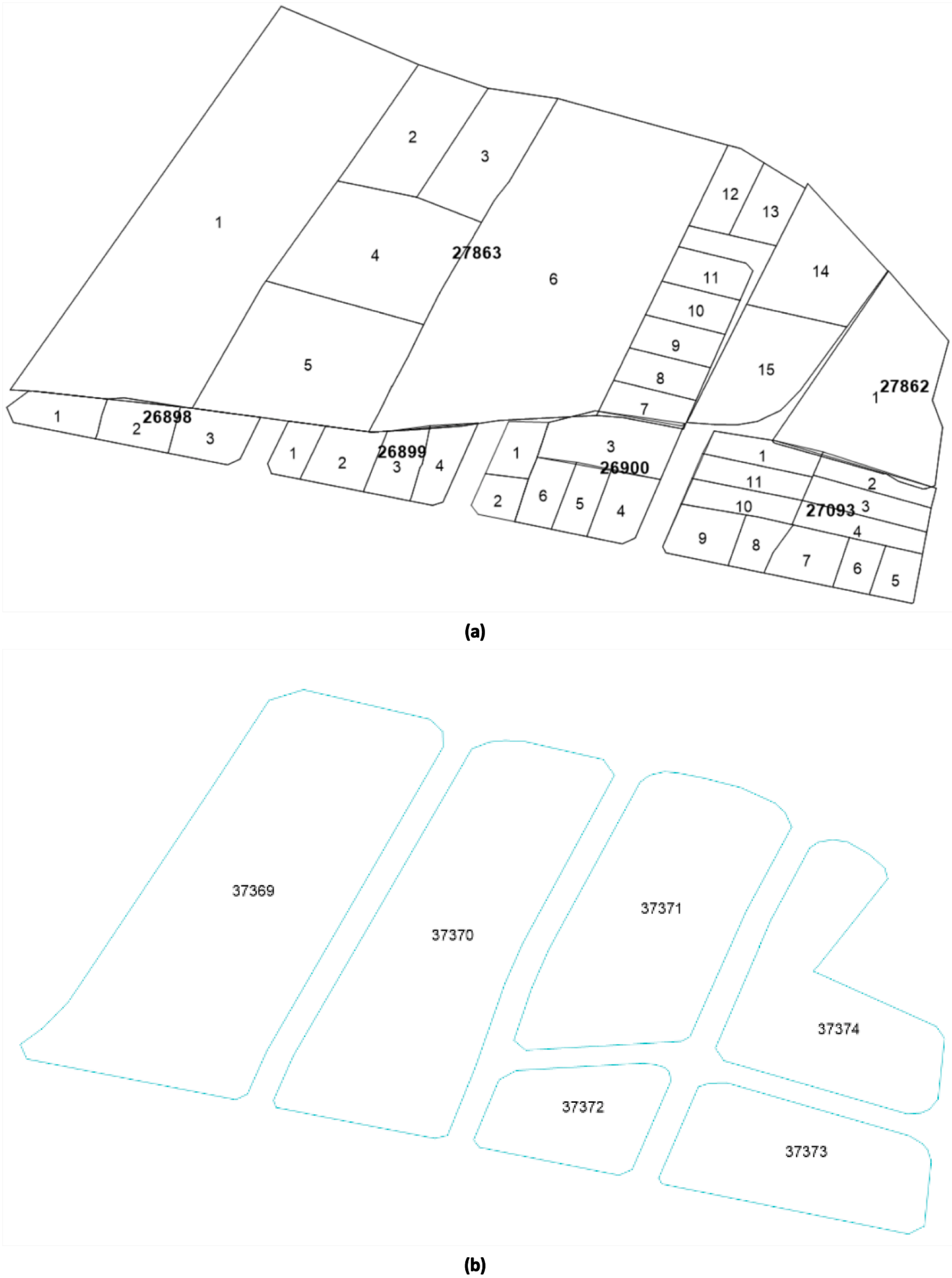


**Fig. 8.** Input data of application area (**a**) Cadastral block map containing parcels (**b**) Zoning block map.

given in Table 5. The results obtained by using formal map output of that the technician have made manually are added to the table, too. When these results are considered in terms of *Cost*, it can be seen that the PSO method achieves the best result as compared with ABC and DE. The PSO algorithm shows a very superior performance by finding the *Share* value as 0. In addition, PSO is seen to become better than ABC and DE algorithms in terms of *Gap* and *Distance values*. When considering the NS/NP ratio, the PSO algorithm again shows the highest success. This ratio shows the number of shareholders per parcel. In this paper, in order to make the map outputs obtained from ABC, DE and PSO algorithms directly available in the real world, these maps are given directly to an expert system (ES) and final results are immediately obtained by ES.

**Table 2**
All the parameters of the algorithms used in experimental studies.

| ABC | | DE | | PSO | |
|---|---|---|---|---|---|
| Parameter Name | Value | Parameter Name | Value | Parameter Name | Value |
| Population Size | 20 | Population Size | 20 | Population Size | 20 |
| Number of Iteration | 1000 | Number of Iteration | 1000 | Number of Iteration | 1000 |
| MaxFEs | 20,000 | MaxFEs | 20,000 | MaxFEs | 20,000 |
| Limit Value | 50 | CR Rate (between r2 and r3) | 0.80 | CR Rate (between current and pBest) | 0.80 |
| Selection Operator | Tournament | CR Rate (between r1 and r2) | 0.90 | CR Rate (between current and Gbest) | 0.90 |
| SubSet Size | 3 | CR Rate (between current and r1) | 0.70 | CR Rate (between pBest and Gbest) | 0.70 |
| CR Rate (Employee phase) | 0.70 | | | | |
| CR Rate (Onlooker phase) | 0.90 | | | | |

**Table 3**
Comparative results of the metaheuristic algorithms in terms of performance.

| | | Max Cost | Min Cost | Avg Cost | Std Cost | Max Cost By ES | Min Cost By ES | Avg Cost By ES | Std Cost By ES |
|---|---|---|---|---|---|---|---|---|---|
| *ABC* | *CPC* | 17.388 | 14.447 | 15.770 | 0.762 | 20.203 | 17.640 | 18.563 | 0.683 |
| | *IPC* | 9.634 | 7.460 | 8.654 | 0.615 | 13.849 | 9.362 | 11.644 | 1.031 |
| *DE* | *CPC* | 17.766 | 16.173 | 16.947 | 0.470 | 21.230 | 17.750 | 19.359 | 0.741 |
| | *IPC* | 8.428 | 6.479 | 7.406 | 0.494 | 11.692 | 9.147 | 10.523 | **0.726** |
| *PSO* | *CPC* | 11.717 | 9.432 | 10.375 | 0.573 | 15.953 | 12.288 | 14.089 | 0.930 |
| | *IPC* | **5.817** | **3.929** | **4.947** | **0.449** | **9.687** | **6.652** | **8.050** | 0.841 |


**Fig. 9.** The formal distribution and subdivision outputs obtained by an expert/technician.

When these algorithms, called ABCwES, DEwES and PSOwES, are analyzed, it can be clearly seen that these algorithms obtain much better results than the 23,419 result which is obtained by the technician. PSOwES, on the other hand, reached 8,013 in terms of *Cost* value and reveals a success far above the value that the technician achieved. Besides, when the results obtained by PSOwES method are examined, it can be seen that this method is much more successful than technician's results in terms of *Share* and *Distance* values. It is clear that the 0.014% (~48,000x0.0014 = ~ 6.72 m$^2$) value attained with PSOwES is a very small and negligible area value, although the technician result seems to be better in terms of *Gap value*. Therefore, the solutions obtained with the proposed algorithms can be directly used in the real world without any intermediate process or tool.

Figs. 13 and 14 show the subdivision map of the best results obtained by using the PSO algorithm and PSOwES, respectively. In these maps, which consists of 6 zoning islands, the angle of the adjacent edge of the parcels with the next parcel is automatically performed in parallel with the edge angle of each block. This leads to the creation of a regular

**Table 4**
Comparative results of the metaheuristic algorithms in terms of time.

| | | Max Time (min) | Min Time (min) | Avg Time (min) | Std Time (min) |
|---|---|---|---|---|---|
| *ABC* | *CPC* | **4.558** | **4.220** | **4.335** | **0.087** |
| | *IPC* | 5.196 | 4.603 | 4.949 | 0.128 |
| *DE* | *CPC* | 10.444 | 8.907 | 9.445 | 0.422 |
| | *IPC* | 13.282 | 11.557 | 12.397 | 0.523 |
| *PSO* | *CPC* | 10.427 | 8.535 | 9.298 | 0.475 |
| | *IPC* | 13.878 | 11.941 | 13.026 | 0.511 |

subdivision map that can be used without any need for pre-processing. Fig. 14 provides the ultimate solution. On the other hand, the map given in Fig. 13 can be accepted as an intermediate solution since it contains overflow or space areas in the blocks. As an example of overflow areas in the block area, parcel 7 in block 37,369 can be given. While the area of this parcel is 708.3 $m^2$ as a result of PSO, this parcel's area is updated as 616.2 $m^2$ by LRES. As another example of the gap areas, parcel 10 in block 37,373 can be also given. As a result of PSO, the area of the parcel is 430.6 $m^2$, while this area is updated as 422.9 $m^2$ by LRES.

When comparing Figs. 13 and 14, the areas of parcel numbered 6 and 7 in block 37369, parcel numbered 8 and 9 in block 37370, parcel numbered 4 and 5 in block 37371, parcel numbered 1 and 10 in block 37373, and finally parcel numbered 3, 4, 5 and 12 in block 37,374 are

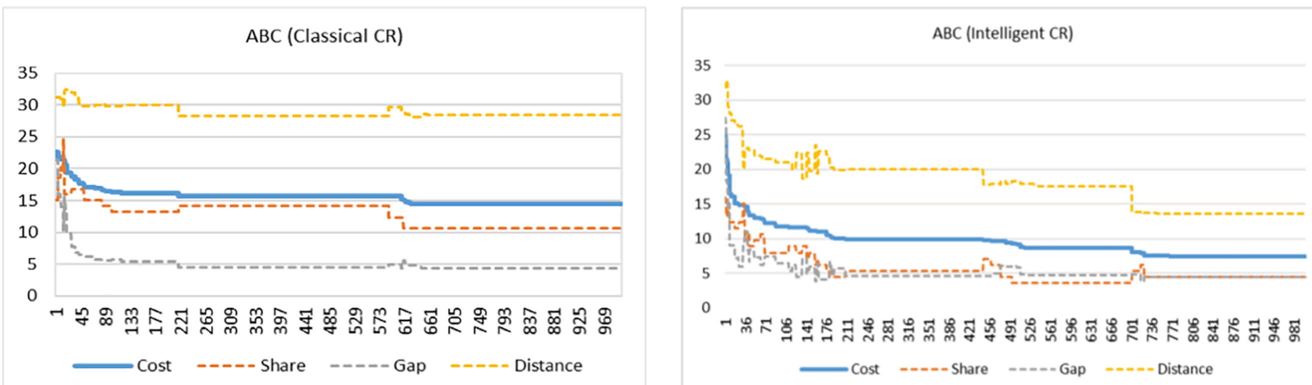


**Fig. 10.** Convergence graphics of ABC algorithm.

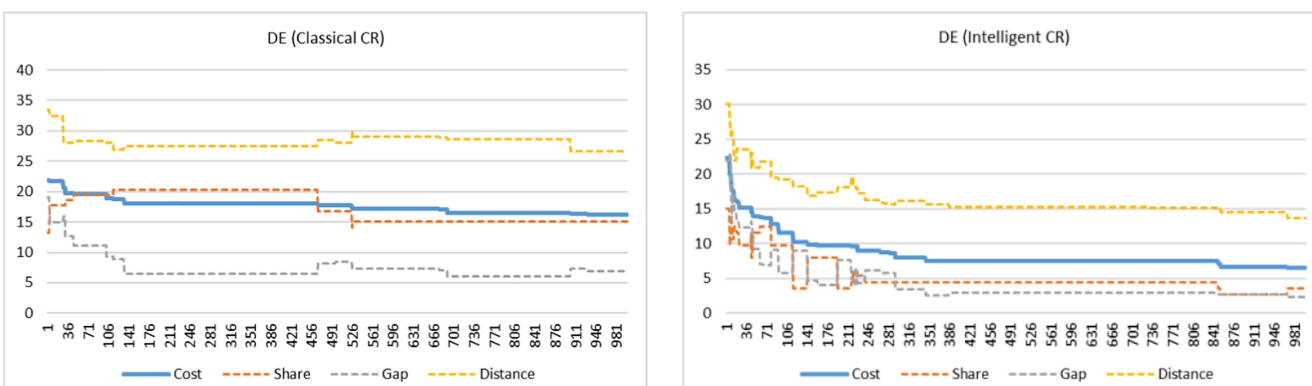


**Fig. 11.** Convergence graphics of DE algorithm.

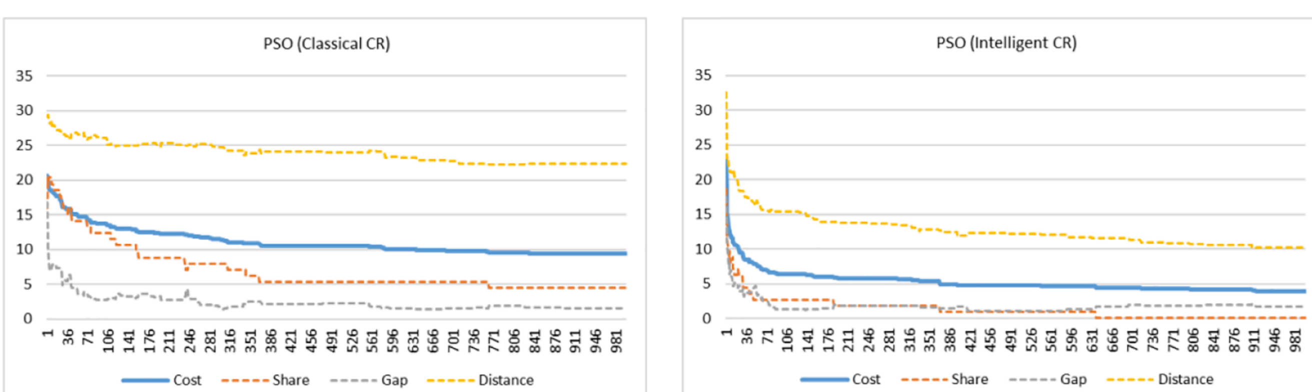


**Fig. 12.** Convergence graphics of PSO algorithm.

**Table 5**
Comparative analysis of different techniques.

| Technique | Cost | Share (%) | Gap (%) | Distance | Number of Parcel (NP) | Number of Shareholder (NS) | NS/NP |
|---|---|---|---|---|---|---|---|
| *ABC* | 7.460 | 4.425 | 4.428 | 13.550 | 63 | 113 | 1.794 |
| *ABCwES* | 9.362 | 12.389 | 0.027 | 15.699 | 62 | 124 | 2.000 |
| *DE* | 6.479 | 3.540 | 2.258 | 13.659 | 62 | 113 | 1.823 |
| *DEwES* | 9.875 | 14.159 | 0.019 | 15.477 | 60 | 123 | 2.050 |
| *PSO* | **3.929** | **0.000** | 1.651 | **10.148** | 64 | 113 | **1.766** |
| *PSOwES* | **8.013** | **11.504** | 0.014 | **12.545** | 63 | 122 | **1.937** |
| *Technician* | 23.419 | 43.363 | **0.002** | 26.892 | 67 | 195 | 2.910 |

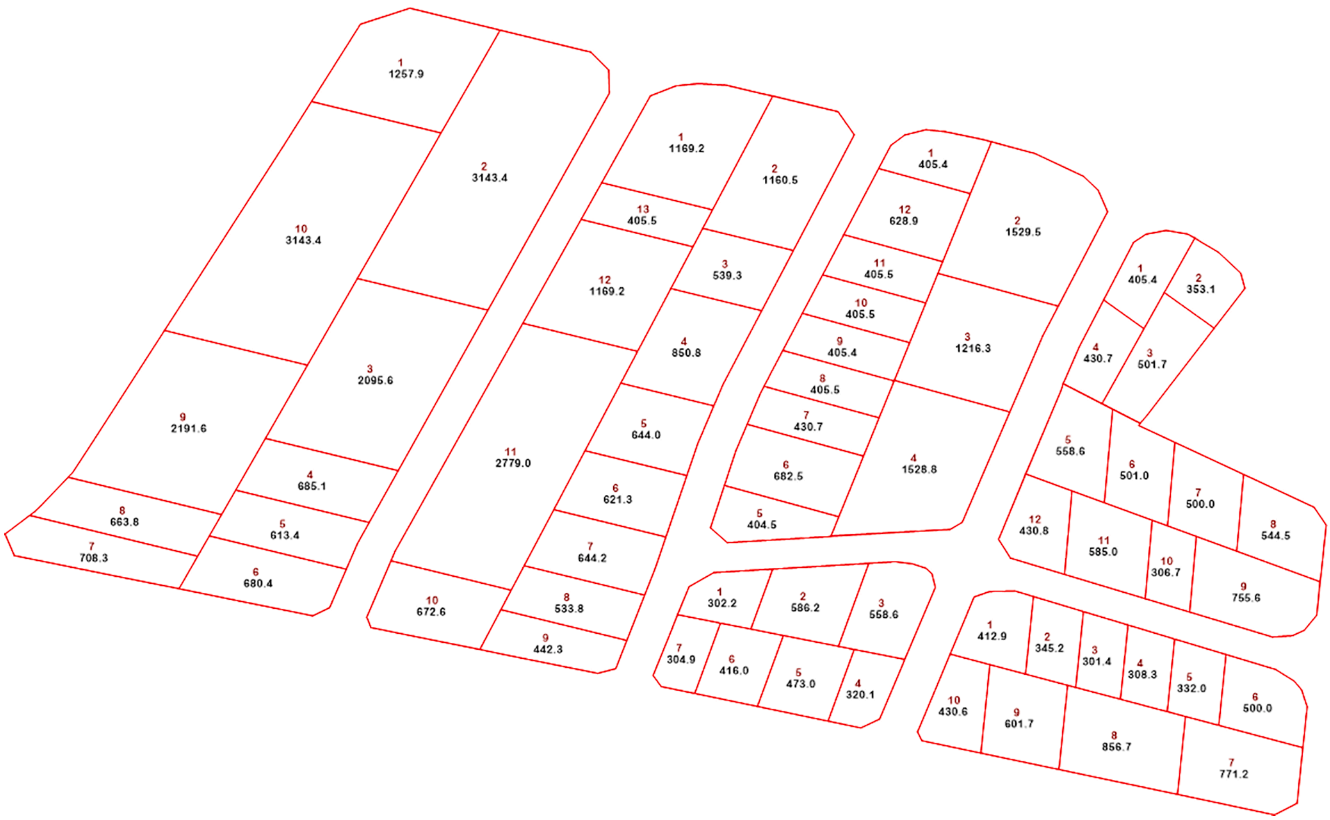


**Fig. 13.** The most optimal reallocation and subdivision results of application area by PSO algorithm.

seen to be updated. Also, when the block numbered 37,372 is examined, it is seen that the total number of parcels changed. Block 37,372 consisting of 7 parcels obtained by PSO algorithm, decreases to 6 parcels after the ES operation. In these processes, as described in detail in ES, land reallocation is used to eliminate overflows or gaps in block areas. As a result, when the Figs. 13 and 14 are examined in detail, it is seen that there is no change in the other parcels except only the last parcels in terms of area.

In this study, 3 different penalty points are used in the proposed objective function. Two of these penalty points are specified in Article 18 (Web2, 2020). In addition, *Gap* value are included in the proposed objective function. The advantage of *Gap* penalty point being determined as a goal is that it allows the algorithm to work flexibly. Otherwise, it would be necessary to divide any owner into more shareholders or a particular area of owners as much as the free space at each iteration of the algorithm. This would increase the number of shareholders. Hence, this is important for the convergence to the optima of the algorithm. When analyzing results, PSO is clearly seen to be the best algorithm as compared with ABC and DE in terms of all the 3 objectives. When the results obtained by PSO algorithm are examined, it can be seen that *Share* value is 0. However, *Distance* value is the worst penalty value in all the solutions. The reason for this is that zoning and cadastral blocks do not overlap exactly with each other shown in Fig. 9. It is obvious that the *Gap* value will remain at a certain level even if the algorithm converges to the most optimal value due to the fact that the sum of areas of any subset of the randomly selected owners are not completely equal to the block areas in this project. Therefore, it is obvious that *Gap* and *Distance* values will never be 0 for this project area. As a result, it can be said that the PSO method is more robust and more successful than other algorithms. It is obvious that IPC technique is much more successful than CPC. It can be clearly seen from the experimental results that the developed expert system produces an effective solution by making very small changes on the existing solutions acquired by the algorithms. Finally, it can be clearly seen from the experimental results that the obtained solutions have higher quality than that of the technician, and the algorithms are much faster than the expert in terms of time.

### 5.1. *Discussion*

In the experimental studies, three different metaheuristic algorithms including ABC, PSO and DE are used to solve the LR problems. The implementations of these algorithms to the LR problem are presented in detail in Fig. in 11, 12 and 13. The LR problems are discrete optimization

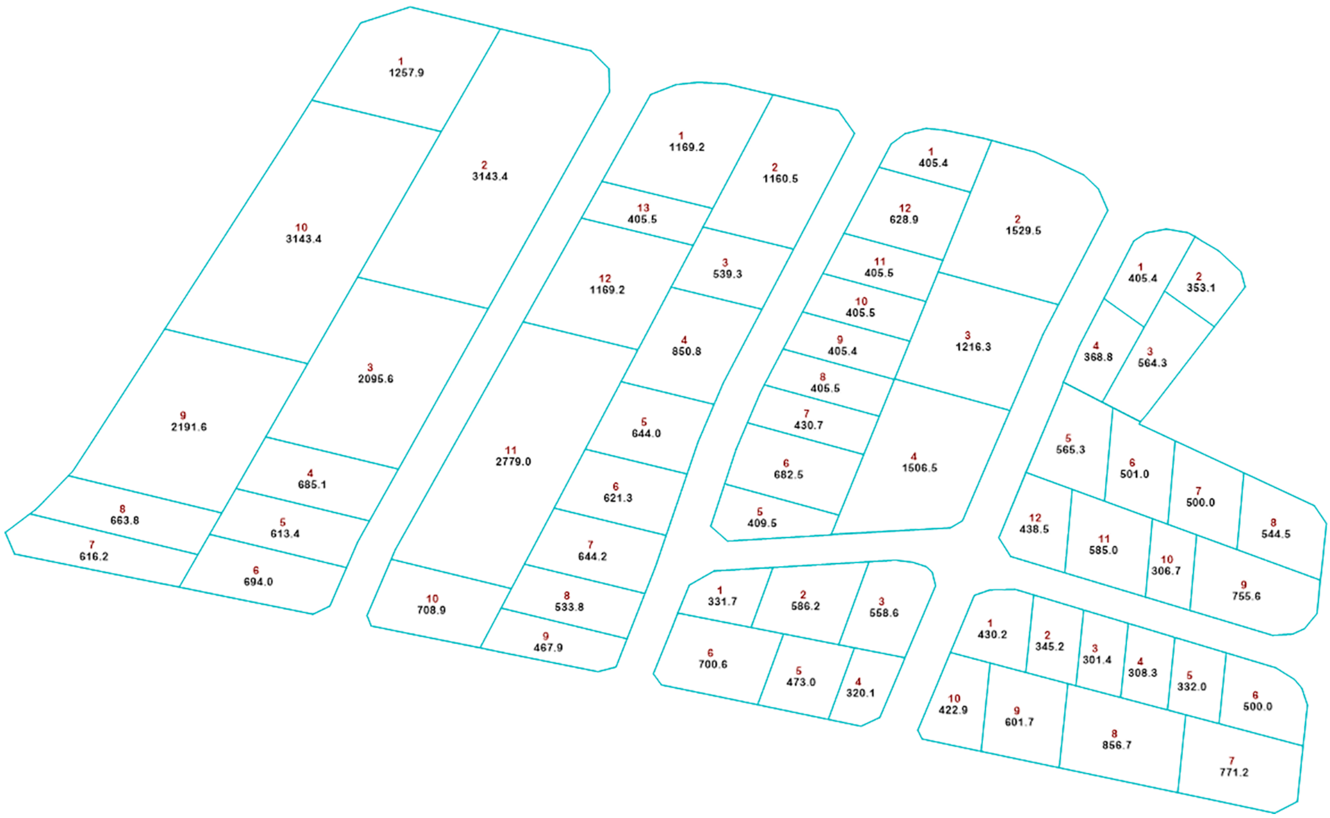


**Fig. 14.** The final reallocation and subdivision results of application area by PSOwES.

problems, and has its own constraints. In this study, considering the side lengths of the parcel, the smallest area of a parcel is determined as 300 $m^2$. This situation changes depending on the width of the zoning blocks (facade length constraint). For example, in Block 37369, the minimum parcel area is calculated as 600 $m^2$ according to the width of the block. Technical details are presented in the studies of (Koc, et al., 2020). In addition, a middle line is manually determined in each block to easily calculate the boundaries of the parcels with the binary search algorithm (BS). Thanks to this line, the new boundaries of the parcels in both sides of the blocks are more easily determined according to the assignment status of the owners to the parcels. For this process, the total area of the corresponding parcel is determined according to the owner lists assigned to the parcels (Total parcel area = sum of the areas of the owners in the corresponding parcel). Then, the boundaries and center point of each parcel are determined from north to south (for vertical blocks) and from east to west (for horizontal blocks) using the BS method. According to these center points, distance values in objective function are used.

In this study, a new approach proposed for solving LR problems is presented by utilizing metaheuristic algorithms on a real project data. In this article, ABC, DE and PSO algorithms are used, and it is experimentally observed that the PSO algorithm produced superior results. Considering that the local search capability of the PSO algorithm is better than those of ABC and DE algorithms, it can be said that algorithms with stronger local search such as PSO can be more effective on real project data. Especially due to the non-uniform blocks, the different side lengths of the blocks and the very different share areas of the owners on a real project site, even small changes on the candidate solution obtained can lead to divergence from the optimal result. For example, in case of swapping any selected two owners between any two blocks, the overflow or gap rates in the block areas can change significantly, since the areas of the owners are different from each other and even vary greatly. Performing this operation with many more owners would negatively affect the distance, share and overflow/space criteria in the objective function. Because of all these situations, it is recommended to prefer algorithms with stronger local search for solving LR problems in real project sites.

The algorithms used in this study use two new parcel-based approaches as crossover and mutation operators, unlike existing operators in the literature. A hybrid approach is proposed by combining these crossover and mutation operators, and this approach is given in Fig. 7. In this method, first a crossover operator from two crossover operators is determined by the user, and then two different mutation operators are used respectively. The updated parcel boundaries and centers of the parcels are calculated by BS algorithm. In this way, a new candidate solution is generated. As for the proposed crossover operators, two different crossover operators are implemented to the algorithms, the random approach (CPC) and the greedy approach (IPC). In the CPC method, all owner replacement operations take place if the random condition is met, while in the IPC method, these operations occur if the new situation is better than the previous situation. This is an approach that always guarantees a good solution. It is clearly seen in the experimental results that in all the algorithms, IPC produces much better results than CPC and its convergence continues. According to these results, the IPC method can be promisingly accepted as an alternative crossover operator for the real project site.

Using the proposed hybrid approach, metaheuristic algorithms obtain the final solutions after a certain iteration. Although these solutions are close to the ideal solution, they do not exactly overlap with the block areas in the current project site. In other words, there are overflows or empty areas on the blocks of the solutions produced. Therefore, these solutions cannot be applied directly to a real world. In order to handle this problem, a new expert system (LRES) is designed specifically for LR problems. Thanks to LRES, overflow and empty areas are detected and the owners in these areas are replaced completely or with certain shares. This process is continued until the overflow and gap areas in the blocks are completely eliminated. Thus, suitable solutions obtained from algorithms are made completely feasible in real life.

## 6. Conclusion and future works

In this paper, artificial bee colony, particle swarm optimization and differential evolutionary algorithms are used for solving LR problems. For performing the LR practices, two main objectives are determined according to 18th Article (Web2, 2020). In accordance with the nature of the problem, two different crossover operators are proposed named as Classical and Intelligent Parcel-based Crossover operators, shortly, CPC and IPC, respectively. Especially thanks to IPC, the algorithms present very effective results. The obtained results are given directly to the developed expert system, and overflow and gap areas in the blocks are completely removed instantly. Thus, distribution and subdivision plans produced by this hybrid method can be used directly in the real world. When the final solutions obtained from the algorithms supported by the expert system are compared with the solution produced manually by the expert/technician, it can be seen that the proposed methods are clearly very effective. In addition, among the evaluated metaheuristic algorithms, PSO is clearly seen to be very successful in terms of performance than ABC and DE algorithms. When evaluated in terms of time, while an expert can obtain plans manually in an average of 3 days for this project area, the proposed methods offer very high-quality solutions which can be accomplished between 5 and 10 min. In case the project area grows, the contribution of the proposed methods in terms of time will become clearer.

With the metaheuristic algorithms on the real project site, much better solutions are produced in a very short time than the manual solution. It can be clearly seen that the PSO approach is particularly effective, and it can be deduced that the basis of this success is the superiority in local search capability. Based on this inference, algorithms with high local search capability can be adapted to LR problems in order to produce more effective outputs (reallocation and subdivision plan). In this sense, the point to be considered is that if any algorithm works in continuous space, all the steps of the algorithm should be designed to work in a discrete space. This process should be done starting from the initial population phase. After this phase, the hybrid approach, which includes the crossover and mutation operators specially designed for LR problems, is used throughout the iteration process in the new solution generation process of the algorithm. In summary, it can be said that algorithms with stronger local search capability would produce better results than algorithms with strong global search capability for real project data.

In this study, two different crossover operators are proposed namely CPC and IPC. Among these operators it can be clearly seen that IPC operator produces much more effective results rather than CPC. The use of the IPC operator within the developed algorithms for LR problems would seriously affect the result. In short, an intelligent based crossover operator rather than random-based approach seems to be very helpful in producing effective results, especially for real project data. At this point, when the project area is expanding, it is thought that IPC would be much more successful than the CPC considering that the convergence power of the IPC operator.

In this study, an objective function is proposed that simultaneously evaluates three criteria (share, distance and gap ratio) specific to LR problems. The success of this objective function is observed from the convergence curves plotted for each criterion. According to these curves, it can be seen that each criterion continues to converge within itself during the iterations and they do not show superiority to each other. Hence, using this form of the objective function, algorithms can directly generate efficient solutions to LR problems. In addition, by changing the coefficients of each criteria in the objective function, analyzes can be carried out, and a more effective objective function can be produced accordingly. Furthermore, a new objective function can be produced by different multi-objective optimization techniques.

With the help of the expert system (LRES) produced for the LR problems, the gaps and overflowing areas on the blocks of the solutions obtained from the algorithms are eliminated. This process is done by sorting the overflow and gap areas from largest to smallest, and then matching the largest overflow area to the largest gap area. In this way, this process is carried out by changing the position of the minimum number of owners, and there is a little change between the solution after LRES and the solution produced by the algorithm. For future studies, instead of this approach, a stronger expert system can be designed by taking into account the distance and share status of the owners that are planned to be replaced.

## Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgements

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## References

Aggarwal, D., Saxena, D. K., Back, T., & Emmerich, M. (2020). Real-World Airline Crew Pairing Optimization: Customized Genetic Algorithm versus Column Generation Method. *arXiv preprint arXiv:2003.03792*.
Ali, M., Pant, M., & Abraham, A. (2009). Simplex differential evolution. *Acta Polytechnica Hungarica, 6*, 95–115.
Bansal, J. C., Singh, P., Saraswat, M., Verma, A., Jadon, S. S., & Abraham, A. (2011). Inertia weight strategies in particle swarm optimization. In *2011 Third world congress on nature and biologically inspired computing* (pp. 633–640). IEEE.
Chou, T., & Shen, S. (1982). Urban land readjustment in Kaohsiung, Taiwan. *Land readjustment: A different approach to financing urbanization. Mass., Lexington Books, USA*.
Colak, H. E., & Memisoglu, T. (2021). Evaluating success factors in the land readjustment applications: A case study of Trabzon Province. *Turkey. Survey Review, 53*(376), 70–79.
Coruhlu, Y. E., Uzun, B., & Yildiz, O. (2020). Zoning plan-based legal confiscation without expropriation in Turkey in light of ECHR decisions. *Land use policy, 95*, 104598. https://doi.org/10.1016/j.landusepol.2020.104598
Çağdaş, V., & Linke, H. J. (2020). An institutional analysis of land readjustment in Turkey. *Survey Review, 53*(378), 252–262.
Çete, M. (2010). Turkish land readjustment: Good practice in urban development. *Journal of urban planning and development, 136*(4), 373–380.
Doebele, W. (1982). Land Readjustment: A Different Approach to Financing Urbanization, Health and Company. In: Lexington Books, Massachusetts, DC.
Hrstka, O., & Kučerová, A. (2004). Improvements of real coded genetic algorithms based on differential operators preventing premature convergence. *Advances in Engineering Software, 35*(3-4), 237–246.
İnam, Ş. (2013). Kentsel alan düzenlemelerinde alternatif uygulama yöntemleri ve sonuçların analizi. *Selçuk Üniversitesi Fen Bilimleri Enstitüsü.*
Karaboga, D. (2005). An idea based on honey bee swarm for numerical optimization. In: Technical report-tr06, Erciyes university, engineering faculty, computer ….
Karaboga, D., & Basturk, B. (2007). A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm. *Journal of global optimization, 39*(3), 459–471.
Karaboga, D., Gorkemli, B., Ozturk, C., & Karaboga, N. (2014). A comprehensive survey: Artificial bee colony (ABC) algorithm and applications. *Artificial Intelligence Review, 42*(1), 21–57.
Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. *Proceedings of ICNN'95-International Conference on Neural Networks* (Vol. 4,, 1942–1948.
Keskintürk, T. (2006). Diferansiyel gelişim algoritması. *İstanbul Ticaret Üniversitesi Fen Bilimleri Dergisi, 5*, 85–99.
Koc, I., & Babaoglu, I. (2021). A comparative study of swarm intelligence and evolutionary algorithms on urban land readjustment problem. *Applied Soft Computing, 99*, 106753. https://doi.org/10.1016/j.asoc.2020.106753
Koc, I., Cay, T., & Babaoglu, I. (2020). Approaches to automated land subdivision using binary search algorithm in zoning applications. In *Proceedings of the Institution of Civil Engineers-Municipal Engineer* (pp. 1–13). Thomas Telford Ltd.
Kokturk, E., & Kokturk, E. (2009). Eşdeğerlik İlkesine Dayalı Arsa Düzenlemesinde Taşınmaz Değerlerinin Belirlenmesi. *Jeodezi ve Jeoinformasyon Dergisi*, 10–17.
Kucukmehmetoglu, M., & Geymen, A. (2016). Optimization models for urban land readjustment practices in Turkey. *Habitat international, 53*, 517–533.
Marinakis, Y., Marinaki, M., & Matsatsinis, N. (2009). A hybrid discrete artificial bee colony-grasp algorithm for clustering. In *2009 International Conference on Computers & Industrial Engineering* (pp. 548–553). IEEE.
Mayer, D. G., Kinghorn, B. P., & Archer, A. A. (2005). Differential evolution–an easy and efficient evolutionary algorithm for model optimisation. *Agricultural Systems, 83*(3), 315–328.

Sammut, C., & Webb, G. I. (Eds.). (2010). *Encyclopedia of Machine Learning*. Boston, MA: Springer US.
Storn, R., & Price, K. (1997). Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization, 11*, 341–359.
Turk, S. S. (2005). Land readjustment: An examination of its application in Turkey. *Cities, 22*(1), 29–42.
Turk, S. S. (2007). An analysis on the efficient applicability of the land readjustment (LR) method in Turkey. *Habitat international, 31*(1), 53–64.
Türk, Š.Š. (2009). *Arazi ve arsa düzenlemesi yöntemi ve uluslar arası çerçevede etkin uygulanabilirliği* (p. 8). ITU Journal Series A: Architecture, Planning, Design.
Uygun, U. (2011). *İmar Uygulamasında Değer Esası Kriterlerine Göre Dağıtım Probleminin Yapay Zekâ Yöntemiyle Çözümü*. Yüksek Lisans Tezi: Gebze Yüksek Teknoloji Enstitüsü.
Uzun, B. (1992). *Kentsel Alan Düzenlemelerinde İmar Parseli Üretme Yöntemleri ve Sonuçlarının İrdelenmesi*. Trabzon: Karadeniz Technical University, MS Geodesy And Photometry Engineering.
Uzun, B. (2009). Using land readjustment method as an effective urban land development tool in Turkey. *Survey Review, 41*(311), 57–70.
Web1. (2020). Land and building development law (Imar kanunu). *http://www.mevzuat.gov.tr, (Last access: Apr 29, 2020)*..
Web2. (2020). Regulation of the 18th article of the land and building development law. *http://www.mevzuat.gov.tr, (Last Access: Apr 29,2020)*.
Yilmaz, A., Çağdaş, V., & Demir, H. (2015). An evaluation framework for land readjustment practices. *Land use policy, 44*, 153–168.
Yomralioglu, T. (1992). Arsa ve Arazi Düzenlemesi için Yeni bir Uygulama Şekli.
Yomralioglu, T., Nisanci, R., & Yildirim, V. (2007). An implementation of nominal asset based land readjustment. *FIG Working Week*, 13–17.