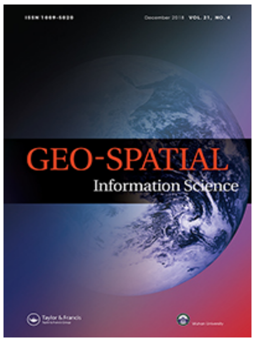

Geo-spatial Information Science

ISSN: 1009-5020 (Print) 1993-5153 (Online) Journal homepage: https://www.tandfonline.com/loi/tgsi20

An improved knowledge-informed NSGA-II for multi-objective land allocation (MOLA)

Mingjie Song & Dongmei Chen

To cite this article: Mingjie Song & Dongmei Chen (2018) An improved knowledge-informed NSGA-II for multi-objective land allocation (MOLA), Geo-spatial Information Science, 21:4, 273-287, DOI: 10.1080/10095020.2018.1489576

To link to this article: https://doi.org/10.1080/10095020.2018.1489576

GEO-SPATIAL INFORMATION SCIENCE
2018, VOL. 21, NO. 4, 273–287
https://doi.org/10.1080/10095020.2018.1489576

OPEN ACCESS

An improved knowledge-informed NSGA-II for multi-objective land allocation (MOLA)

Mingjie Song and Dongmei Chen

Department of Geography and Planning, Queen's University, Kingston, Ontario, Canada

ABSTRACT

Multi-objective land allocation (MOLA) can be regarded as a spatial optimization problem that allocates appropriate use to certain land units subjecting to multiple objectives and constraints. This article develops an improved knowledge-informed non-dominated sorting genetic algorithm II (NSGA-II) for solving the MOLA problem by integrating the patch-based, edge growing/decreasing, neighborhood, and constraint steering rules. By applying both the classical and the knowledge-informed NSGA-II to a simulated planning area of 30 × 30 grid, we find that: when compared to the classical NSGA-II, the knowledge-informed NSGA-II consistently produces solutions much closer to the true Pareto front within shorter computation time without sacrificing the solution diversity; the knowledge-informed NSGA-II is more effective and more efficient in encouraging compact land allocation; the solutions produced by the knowledge-informed have less scattered/isolated land units and provide a good compromise between construction sprawl and conservation land protection. The better performance proves that knowledge-informed NSGA-II is a more reasonable and desirable approach in the planning context.

ARTICLE HISTORY
Received 1 December 2017
Accepted 5 March 2018

KEYWORDS
Multi-objective land allocation (MOLA); non-dominated sorting genetic algorithm II (NSGA-II); knowledge-informed rules

1. Introduction

Multi-objective land allocation (MOLA) can be regarded as a spatial optimization problem that aims to allocate appropriate use to certain land units subjecting to multiple objectives and constraints (Eastman, Jiang, and Toledano 1998; Datta et al. 2007). MOLA shares the common difficulties and complexities of multi-objective optimization problems: the multiple non-commensurable and conflicting objectives have to be optimized simultaneously. There are two popular methods to solve multi-objective optimization problems: the weighted-sum method and the Pareto-based method (Coello, Lamont, and Van Veldhuizen 2007; Duh and Brown 2007; Deb 2014). The weighted-sum method combines the multiple objectives into a single one by multiplying each objective with a user-supplied weight. It requires prior knowledge to assign the weights and its solutions depend greatly on the prior knowledge. This approach can only generate a single solution under a given trade-off scheme between multiple objectives. The Pareto-based optimization methods attempt to find non-dominated solutions, which cannot improve one objective without degrading some others and reflect different trade-offs between multiple objectives. As a spatial optimization problem, MOLA has extra difficulties and complexities, such as the remarkably large data size, the nonlinear objective functions, and the interdependency between spatial variables. Classical optimization methods, such as enumeration, linear programing, branch and bound in integer programing, are found not feasible or appropriate to solve the MOLA problem (Datta et al. 2007; Tong and Murray 2012). Meanwhile, these classical optimization methods cannot solve multi-objective optimization problems following the Pareto-based method; the multiple objectives have to be combined into a single objective function.

Heuristic algorithms, such as simulated annealing (SA), genetic/evolutionary algorithms (GAs/EAs), and particle swarm optimization (PSO) have been developed to produce approximate solutions to solve optimization problems. They have been found applicable to the MOLA problem (Aerts and Heuvelink 2002; Datta et al. 2007; Duh and Brown 2007; Santé-Riveira, Crecente-Maseda, and Miranda-Barrós 2008; Cao et al. 2011, 2012; Masoomi, Mesgari, and Hamrah 2013; Liu et al. 2016; Sahebgharani 2016). However, since heuristic algorithms cannot guarantee optimal solutions, they suffer the risk of being trapped in local optima. Also, because of a large number of possible solutions, and the spatial relationships/patterns that have to be considered in MOLA problems, it usually takes extremely long computation time for these heuristic algorithms to obtain quality solutions. Moreover, these heuristic algorithms were initially developed for single objective optimization problems. Therefore, when the conventional versions of SA, GA, and PSO are applied to solve multi-objective optimization problems, the multiple objectives usually need to be weighted and combined into a single objective function. In recent years, a few modified versions of SA,

CONTACT Dongmei Chen ✉ chendm@queensu.ca

GA, and PSO have been developed particularly for solving multi-objective optimization objectives, such as, non-dominated sorting genetic algorithm (NSGA), Pareto archived evolution strategy (PAES), Pareto SA (PSA), and multi-objective PSO (MOPSO) (Knowles and Corne 2000; Deb et al. 2002; Banks, Vincent, and Anyakoha 2007; Coello, Lamont, and Van Veldhuizen 2007; Duh and Brown 2007; Masoomi, Mesgari, and Hamrah 2013). When these modified algorithms are applied to multi-objective optimization, they can follow both the weighted-sum method and the Pareto-based method.

Since MOLA itself is a knowledge-informed procedure/activity, incorporating knowledge-informed rules into the heuristic optimization algorithms has been regarded as a natural and promising approach to achieve quality solutions for MOLA within a reasonable computation time (Datta et al. 2007; Duh and Brown 2007; Cao et al. 2012; Liu et al. 2016). The knowledge-informed NSGA-II try to avoid redundant and unproductive searches by reducing the size of the search space or changing its structure. Nevertheless, inappropriate knowledge incorporated may also lead the algorithm to converge to sub-optimal solutions.

Duh and Brown (2007) incorporated spatial compactness and contiguity rules into the PSA algorithm to solve a two-objective (minimizing cost and fragmentation) optimization problem in landscape planning. The knowledge-informed rules were applied to decrease landscape fragmentation. Datta et al. (2007) developed a series of knowledge-informed operators for the NSGA-II to solve a three-objective optimization problem in land management, including maximizing economic return, maximizing carbon sequestration, and minimizing soil erosion. The knowledge-informed rules were mainly used for encouraging compact land allocation and steering infeasible solutions to feasible ones. However, using too many operators also increases the computational cost. Cao et al. (2012) designed a few edge-based operators for the NSGA-II for sustainable land use optimization. However, the multiple objectives are combined with weights, and their algorithm only provides a single solution under a given weighting scheme. Their approach fails to reflect the advantages of NSGA-II as a Pareto-based multi-objective optimization technique. Liu et al. (2016) developed a knowledge-informed PSO algorithm for township land use planning in China. Since the knowledge-informed rules applied in the algorithm are problem-specific, it is hard to apply the knowledge-informed operators and algorithms to other MOLA problems.

Among the multiple heuristic optimization algorithms, the NSGA-II has been popularly applied to solve the MOLA problem (Datta et al. 2007; Cao et al. 2011, 2012; Shaygan et al. 2014; Mohammadi, Nastaran, and Sahebgharani 2015). NSGA-II is a modified genetic algorithm (GA) that tries to produce non-dominated solutions for multi-objective optimization problems by simulating the natural selection process. Same as the classical GA, the improvement of solutions is also through the crossover and mutation of solutions in the previous generation. Based on a previous test, NSGA-II has performed better than what SA and PSO algorithm did in solving the MOLA problem (Song and Chen 2018).

In previous studies, the knowledge-informed rules incorporated into the NSGA-II can be categorized into four types: patch-based rules, edge growing/decreasing rules, neighborhood rules, and constraint steering rules (Datta et al. 2007; Cao et al. 2011, 2012; Shaygan et al. 2014; Mohammadi, Nastaran, and Sahebgharani 2015). Based on the patched-based rules, the crossover and mutation are operated on patches instead of randomly selected cells. According to the edge growing/decreasing rules, mutations are operated on the edge cells of patches. Following the neighborhood rules, the land use type of a cell is mutated based on the land use type of its neighboring cells. The above three rules are all used to encourage compact land allocations. Constraint steering rules are used to steering infeasible solutions toward feasible ones. Previous researches usually design multiple operators to incorporate these knowledge-informed rules independently. Fewer studies have tried to design operators that integrate the multiple knowledge-informed rules.

This study aims to develop an improved knowledge-informed NSGA-II by designing operators that integrate the patched-based, edge growing/decreasing, neighborhood, and constraint steering rules to solve the MOLA problem. The knowledge-informed NSGA-II is expected to improve the solution quality without increasing the computation time compared to the classical NSGA-II.

The remainder of the article is organized as follows. Section 2 describes the data and methods with four subsections specifically, including data used for the test, optimization model formulation, the introduction of the classical NSGA-II and our knowledge-informed modifications, and measurements to evaluate the performance of the two algorithms. Section 3 applies both the classical and the knowledge-informed NSGA-II to the MOLA problem, and compares their performance. Section 4 summarizes the research, discusses the methods and results, and identifies future research directions.

2. Data and methods

2.1. *Data*

For the convenience of comparison, this study tests the performance of the algorithms with a semi-hypothetical dataset. A 30 × 30 grid is used as the planning area with physical and socioeconomic characteristics of the land. It is a square area of 9 km^2 in the Xin'andu township of Dongxihu District, Wuhan,

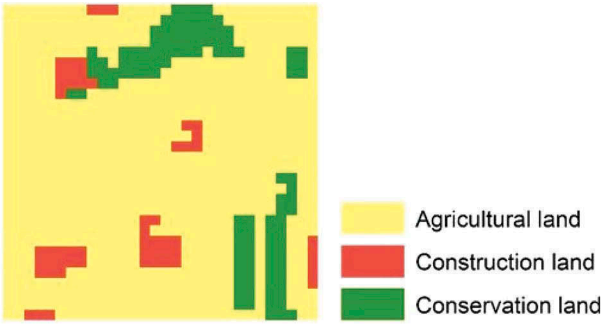

Figure 1. Current land use map.

China (Figure 1). The planning area will be allocated for three land use types, including agricultural land, construction land, and conservation land. The current land use map is shown in Figure 1. It will be used as a benchmark for evaluating the solutions produced by the two algorithms.

The dataset used for the test includes the land suitability data and the current land use data. Land suitability data includes three raster layers representing to what extent each land unit is suitable for agriculture, construction, and conservation use, respectively. They were produced based on a multi-criteria evaluation of physical, socioeconomic, and ecological factors through the analytic hierarchy process (AHP) assisted with the knowledge of local experts and planners (Satty 2008). The land suitability level of each cell for a specfied use is normalized to the range [0,1] through dividing it by the highest suitability level of all cells for the same use in the planning area. Current land use data is used for producing the initial status implemented in the algorithms, and evaluating the solutions.

2.2. Optimization model formulations

There are four objectives and two constraints considered in this optimization model. Their detailed formulations are described as follows.

Objectives:

Objective 1: Maximize agricultural suitability

Maximize $Suit_{agr}$:

$$Suit_{agr} = \sum_{i=1}^{M}\sum_{j=1}^{N} S_{ijagr}x_{ijagr} \tag{1}$$

Objective 2: Maximize construction suitability

Maximize $Suit_{cst}$:

$$Suit_{cst} = \sum_{i=1}^{M}\sum_{j=1}^{N} S_{ijcst}x_{ijcst} \tag{2}$$

Objective 3: Maximize conservation suitability

Maximize $Suit_{csv}$:

$$Suit_{csv} = \sum_{i=1}^{M}\sum_{j=1}^{N} S_{ijcsv}x_{ijcsv} \tag{3}$$

Objective 4: Maximize spatial compactness. It is represented by maximizing the number of cells allocated for the same use in each cell's eight neighboring cells (Aerts et al. 2003; Ligmann-Zielinska, Church, and Jankowski 2008; Cao et al. 2012).

Maximize *Compactness*:

$$Compactness = \sum_{k=1}^{K}\sum_{i=1}^{N}\sum_{j=1}^{M} b_{ijk}x_{ijk} \tag{4}$$

where:

$$b_{ijk} = x_{i-1\,jk} + x_{i+1\,jk} + x_{i\,j-1\,k} + x_{i\,j+1\,k} + x_{i-1j-1\,k} + x_{i-1\,j+1\,k} + x_{i+1\,j-1\,k} + x_{i+1\,j+1\,k} \tag{5}$$

Constraints:

(1) Upper and lower limit for the number of cells allocated for each use

$$A_{kmin} \leq \sum_{i=1}^{M}\sum_{j=1}^{N} x_{ijk} \leq A_{kmax} \tag{6}$$

(1) Current conservation land needs to be protected from agricultural or construction activities.

$$x_{ijcsv} = 1,\ when\ LUC_{ijcsv} = 1 \tag{7}$$

In the above formulas, K is the total number of land use types, and $K = 3$. M and N are the total number of rows and columns of the planning area respectively, and $M = N = 30$. When cell (i, j) is allocated for land use k, x_{ijk}= 1; otherwise, $x_{ijk} = 0$. Specifically, when cell (i, j) is allocated for agriculture, $x_{ijagr} = 1$; otherwise, $x_{ijagr} = 0$. Similarly, $x_{ijcst} = 1$ and $x_{ijcsv} = 1$ represent cell (i, j) is allocated for construction and conservation, respectively. S_{ijagr}, S_{ijcst}, and S_{ijcsv} are the land suitability levels of cell (i, j) used for agriculture, construction, and conservation, respectively. $LUC_{ijcsv} = 1$ means that cell (i, j) is currently used for conservation. $Suit_{agr}$, $Suit_{cst}$, and $Suit_{csv}$ are the agricultural, construction and conservation suitability of the land allocation solution, respectively. Also the objective function values (OFV) of the land allocation solution for Objectives 1, 2, and 3, respectively. *Compactness* is the spatial compactness value. A_{kmin} and A_{kmax} are the lower and higher limits for the number of units allocated for land use k, respectively. Considering the tendency of construction sprawl and the objective of protecting conservation land, we set the target number of cells allocated for agricultural land, construction land and conservation land as 650–694, 83–110, and 108–140, respectively.

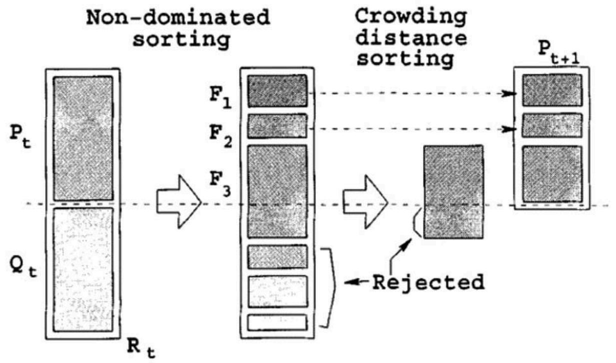

Figure 2. The procedure of NSGA-II.

2.3. *NSGA-II and knowledge-informed modification*

2.3.1. *Overview of NSGA-II*

The NSGA-II is a multi-objective optimization algorithm that tries to identify non-dominated solutions that represent different trade-offs between multiple objectives (Deb et al. 2002). It is a modified GA that simulates the natural selection procedure. In the classical GA, at each generation, the individuals are randomly selected from the current population to reproduce children for the next generation through crossover and mutation. NSGA-II differs from the classical GA by using an elitist principle, an explicit diversity preserving mechanism, and by emphasizing non-dominated solutions. Its procedure is shown in Figure 2. At each generation, NSGA-II selects individuals based on their non-dominated ranks and crowding distances.

The fast-non-dominated-sorting procedure and the crowding-distance-sorting procedure are the key characteristics of NSGA-II. When NSGA-II is applied to solve the MOLA problem, each individual represents one land allocation solution. Crossover exchanges the land use type of selected cells between two land allocation solutions. Mutation flips or changes the land use type of selected cells in one land allocation solution. In our test, we apply the two-point-crossover and swap-mutation operators for the classical NSGA-II, since they have been proven to work well based on the previous test (Song and Chen 2018).

The pseudo code of the fast-non-dominated-sorting procedure is as follows:

```
fast-nondominated-sort($\boldsymbol{P}$)

for each $p \in \boldsymbol{P}$
   for each $q \in \boldsymbol{P}$
       if ($p$ dom($q$)) then             if $p$ dominates $q$ then
           $\boldsymbol{S}_p = \boldsymbol{S}_p \cup \{q\}$                      include $q$ in set $\boldsymbol{S}_p$
       elseif ($q$ dom($p$))   then       if $p$ is dominated by $q$ then
           $n_p = n_p + 1$                         increment $n_p$
   if $n_p = 0$ then                       if no solution dominates $p$ then
      $\boldsymbol{F}_i = \boldsymbol{F}_i \cup \{p\}$                        $p$ is a member of the first front
$i = 1$
while $\boldsymbol{F}_i \neq \varphi$
 $\boldsymbol{H} = \varphi$
 for each $p \in \boldsymbol{F}_i$                     for each memberp in $\boldsymbol{F}_i$
     for each $q \in \boldsymbol{S}_p$                 modify each member from the set $\boldsymbol{S}_p$
         $n_q = n_q - 1$                      decrement $n_q$ by one
         if $n_q = 0$ then $H = H \cup \{q\}$   if $n_q$ is zero, $q$ is a memberof a list $\boldsymbol{H}$
 $i = i + 1$
 $\boldsymbol{F}_i = H$                              current front is formed with all members of $\boldsymbol{H}$
```

The pseudo code of the crowding-distance-sorting procedure is as follows:

```
crowding-distance-assignment($\boldsymbol{L}$)

 $l = [\boldsymbol{L}]$    number of solutionsinI
for each $i$, set $\boldsymbol{L}[i]_{distance} = 0$    initialize distance
for each objective $m$
  $\boldsymbol{L} = \mathrm{sort}(\boldsymbol{L}, m)$                    sort using each objective value
  $\boldsymbol{L}[l]_{distance} = \boldsymbol{L}[l]_{distance} = \infty$    so that boundary points are always selected
  For $i$ = 2 to ($l$-1)                  for all other points
  $\boldsymbol{L}[i]_{distance} = \boldsymbol{L}[i]_{distance} + (\boldsymbol{L}[i+1].m - \boldsymbol{L}[i-1].m)$
```

In classical NSGA-II, the non-dominated solutions are achieved at every generation, and the archive will be updated with new non-dominated solutions in the next generation, while the dominated solutions will be removed from the archive. The final archive contains all non-dominated solutions ever found in the evolution. Under many situations, the number of solutions in the archive will become very large with the increasing generations, which means longer computation time has to be taken to update the archive.

2.3.2. *The knowledge-informed NSGA-II*

We make three aspects of modifications to the classical NSGA-II, including limiting the size of archived solutions, applying knowledge-informed initial solutions, and developing knowledge-informed crossover and mutation operators.

(1) Archive size limitation

To decrease the size while preserving the diversity of the non-dominated solutions, we adopt a truncation mechanism to the archived solutions. The largest number of archived solutions is set equal to the population size. In each generation, among all the non-dominated solutions, the solutions with higher crowding distance will be kept in the archive. The other solutions will be removed from the archive. Since the crowding distance of each solution has been calculated in the crowding-distance-sorting procedure, this solution truncation procedure does not take much extra computation efforts.

(2) Knowledge-informed initialization

Based on the target number of units allocated for each land use, as well as the planning context, many cells will keep their current land use types in the MOLA. Moreover, we need to protect conservation land from agricultural and construction activities. Therefore, instead of using random land allocation patterns as initial solutions, we use initial solutions that keep 70% of current land use, and randomly mutate 30% of current use. Among the 30% cells selected for mutation, if a cell is currently used for conservation, its land use type will not be mutated. This method protects current conservation land from being used for agricultural or construction activities. Based on a comparison of the current number of cells for each land use and the target number of cells for each land use in the MOLA, 30% cells are selected for mutation. This initialization method will be applied to both the classical and the knowledge-informed NSGA-II algorithms.

(3) Knowledge-informed operators

We design three knowledge-informed operators, edge-crossover, patch-based-mutation, and constraint-edge-mutation. Both the edge-crossover and constraint-edge-mutation incorporate the edge growing/decreasing rule, so we first clarify the edge cells here. An edge cell is defined as the cell on the edge of a patch. If a cell's neighboring cells are all used for the same land use with the cell itself, it will not be regarded as an edge cell; otherwise, it is an edge cell. For instance, in Figure 3, R1, R3, and R4 are edge cells, while R2 is not an edge cell.

Based on previous research, the classical NSGA-II is effective and efficient in optimizing non-spatial objectives in MOLA problems, such as minimizing land conversion cost, or maximizing land suitability; but inefficient in handling spatial patterns, such as maximizing spatial compactness/contiguity or preventing land units with specific attributes from certain use (Datta et al. 2007; Cao et al. 2011; Song and Chen 2018). Therefore, the knowledge-informed designations in this research focus on the objective of maximizing spatial compactness and handling the constraints. First, according to the knowledge in planning and landscape ecology, within a land use patch, the core area is more stable due to the homogeneous surroundings, whereas the edge area is volatile; land use change is more likely to happen on the edge area of a patch (Cao et al. 2012; Liu et al. 2016). Based on this knowledge, the operators of edge-crossover, patch-based-mutation, and constraint-edge-mutation encourage compact land allocation by incorporating the edge growing/decreasing, patch-based and neighborhood rules. Specifically, the edge-crossover exchanges the land use type of cells on the edge of two patches; the patch-based-mutation mutates the land use type of patches instead of individual or randomly selected cells, and considers the diverse shape of patches; the constraint-edge-mutation mutates the land use type of edge cells. Second, to deal with the constraints of upper and lower limits for the number of units allocated for each use, the constraint-edge-mutation steers infeasible solutions to feasible ones by comparing the number of units allocated for each use in current

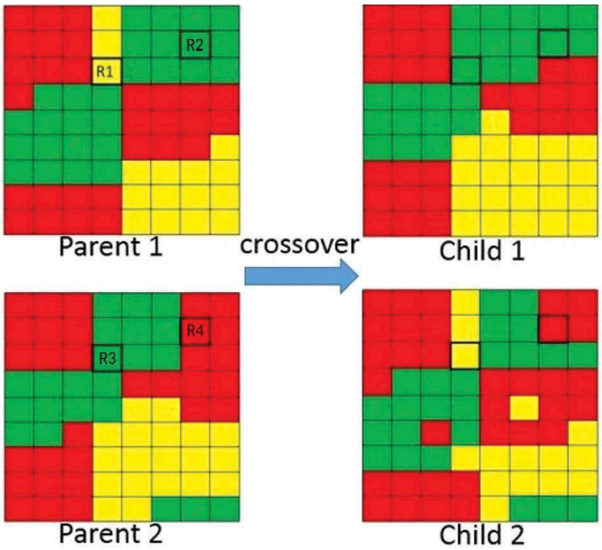

Figure 3. An example of edge-crossover.

solutions to the upper and lower limits. Third, in response to the constraint of protecting current conservation land from agricultural and construction activities, the initial solutions keep all current conservation land, and their land use type is set unchanged in all three operators.

The following part illustrates how the three operators work in detail.

(1) The edge-crossover operator is designed based on the edge growing/decreasing rule. As shown in Figure 3, *Parent* 1 and *Parent* 2 are two randomly selected individuals. To create *Child* 1 and *Child* 2, all the edge cells in *Parent* 1 exchange their land use types with the corresponding cells in *Parent* 2; the other cells keep their original land use type. The pseudo code describes how the edge-crossover works.

```
Pseudo code
Randomly select Parent 1 and Parent 2 to reproduce Child 1 and Child 2.
For p1 in Parent 1, p2 in Parent 2, c1 in Child 1 and c2 in Child 2:
    If p1 is an edge cell:
       c1 = p2, c2 = p1
    Else:
       c1 = p1, c2 = p2
Return Child 1, Child 2
```

(2) The patch-based mutation operator integrates the neighborhood, patch-based, and constraint steering rules. Figure 4 includes an example, and the pseudo code that describe how the patch-based-mutation works.

```
Pseudo code
Step 1: Randomly select a 3 × 3 window.
Step 2: Select 7 cells out of the 9 cells in the window, and regard them as a patch.
Step 3: Determine the most common land use of the 7 selected cells and set it as the land use type of the patch.
Step 4: Mutate the land use types of the 7 selected cells.
   For a cell in the 7 selected cells:
       If the cell is currently used for conservation land:
          The cell keeps its original land use type.
       Else:
          Make the cell's land use type same to the land use type of the patch.
```

(3) Constraint-edge-mutation operator integrates the constraint steering and edge growing/decreasing rules. The parameters used in the operator include the upper and lower limits for the number of cells allocated for each land use. The following is the pseudo code of constraint-edge-mutation, *max_k* and *min_k* represent the upper and lower limits for the number of cells allocated for land use k, respectively.

```
Pseudo code
Step 1: Randomly select an individual.
Step 2: For the individual, calculate the number of cells allocated for each land use, e.g., the number of cells allocated for land use k is num_k.
Step 3: Determine all edge cells.
Step 4:
Randomly select an edge cell i.
  If cell i's current land use is conservation.
     It will still be allocated for conservation land.
  Else:
     If the cell i is allocated for land use k.
        If num_k < min_k:
           Cell i is still allocated for land use k.
        Else if num_k > max_k:
           Cell i is randomly allocated for any land use type other than k.
        Else:
           Cell i is randomly allocated for one of the three land use types.
```

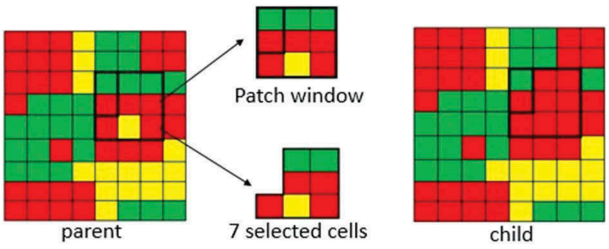

Figure 4. An example of patch-based-mutation.

2.4. *Measurements to evaluate algorithm performance*

2.4.1. *Common criteria to evaluate multi-objective optimization methods*

The Pareto-based multi-objective optimization methods produce a Pareto set/front, which includes non-dominated solutions that cannot improve one objective without degrading some others. The following example explains the non-dominated solutions more specifically. For example, method *A* and method *B* generate solutions *a* and *b* for a multi-objective optimization problem, respectively. If *a* is not worse than *b* as for all objectives, and meanwhile *a* is strictly better than *b* as for at least one objective, then *a* dominates *b*. Otherwise, if *a* is better than *b* as for one or more objectives, but meanwhile *a* is worse than *b* as for other objective(s), then *a* and *b* are non-dominated solutions. The true Pareto front/set is defined as the set/front which contains all solutions that are not dominated by any other solutions. To evaluate the performance of a Pareto-based multi-optimization technique, two criteria are usually considered: the distance of the Pareto set to the true Pareto set, and the diversity of solutions in the Pareto set (Coello, Lamont, and Van Veldhuizen 2007; Duh and Brown 2007). We use a hypothetical two-objective optimization problem to illustrate how these two criteria work.

In Figure 5(a), the Pareto sets generated by the two methods have similar spread, it indicates their solution diversities are similar. However, the Pareto set of method *A* is much closer to the true Pareto set. Under this situation, method *A* performs better than method *B*. In Figure 5(b), the Pareto sets of the two methods have the similar distance to the true Pareto set, but the solutions in the Pareto set of method *B* are more spreading. Under this situation, method *B* performs better than method *A*, since it produces solutions with higher diversity.

2.4.2. *Measurements applied in this study*

To compare the performance of the classical and the knowledge-informed NSGA-II in solving the MOLA problem, besides the two common criteria, we also consider the OFVs of solutions, the computation time, and the mapping pattern of solutions.

In the MOLA problem, the true Pareto set is unknown. The multiple objectives also make it impossible to visually interpret the distance and the diversity of the solutions based on plotting. Therefore, the average rank index (ARI) and the average crowding distance (ACD) of the solutions in the Pareto set are used for evaluating their relative closeness to the true Pareto set and the solution

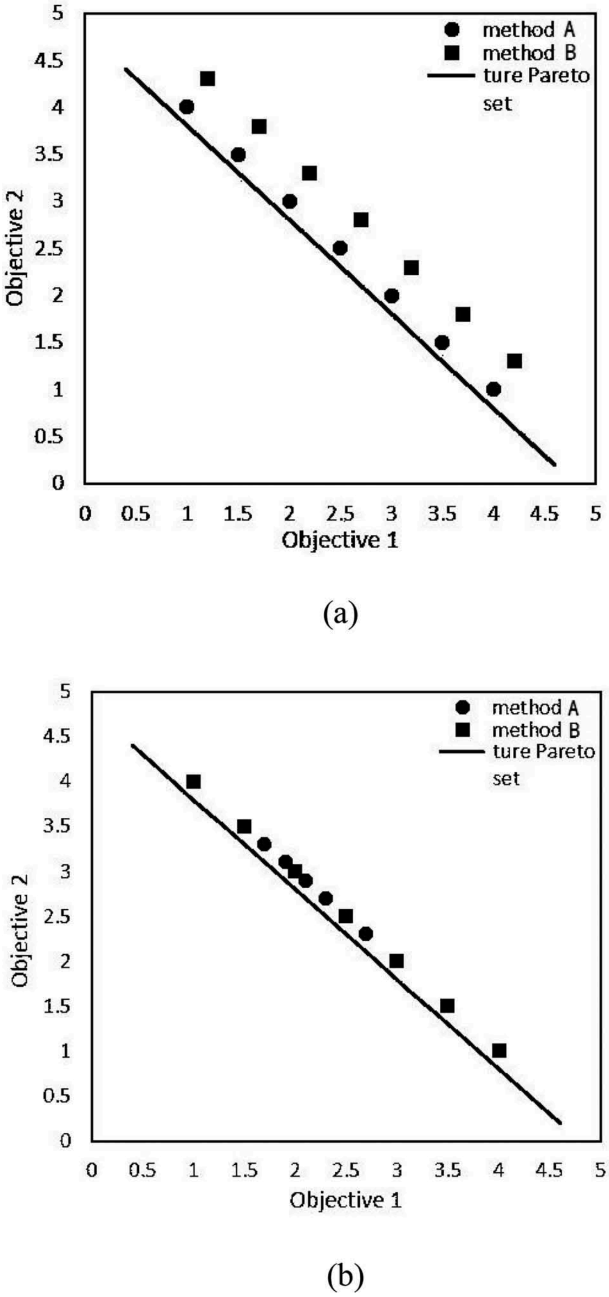

Figure 5. Hypothetical two-objective optimization problem: (a) When Pareto sets generated by two methods have similar spread, it indicates the solution diversities of the two methods are similar. (b) When Pareto sets of two methods have the similar distance to the true Pareto set, the one with more spreading produces solutions with higher diversity.

diversity, respectively (Coello, Lamont, and Van Veldhuizen 2007; Duh and Brown 2007).

ARI: Because the true Pareto front is unknown, it is impossible to calculate the real distance of the solutions to the true Pareto front. ARI is used to evaluate the relative closeness of the solutions to the true Pareto front. The Pareto sets produced by the classical and the knowledge-informed NSGA-II are first pooled and each solution is assigned a Pareto rank based on the fast-non-dominated-sorting-procedure. The procedure sorts all solutions based on their Pareto ranks. The solutions with the Pareto rank 1 are not dominated by any other solutions. The solutions with the Pareto rank 2 are only dominated by the solutions with the rank 1; the lower the rank value is, the closer the solutions are to the true Pareto set. The ranks of all solutions produced by a given algorithm are then averaged to calculate the ARI of the algorithm. For a given algorithm, if its ARI equals to 1, it means that all its solutions are not dominated by any solution produced by the other algorithm. The closer the ARI value is to 1, the closer the corresponding Pareto set is to the true Pareto set. The ARIs of the final Pareto sets and some intermediate Pareto sets are calculated, respectively.

ACD: It is used to evaluate solution diversity; and a larger ACD value indicates higher solution diversity. For each solution in the Pareto set, its crowding distance will be calculated based on the crowding-distance-sorting procedure. The crowding distance of all solutions in the Pareto set will be averaged to obtain the ACD. Since we apply an archive limitation mechanism to the knowledge-informed NSGA-II, the number of solutions in the Pareto set produced by it is less or equal to the population size, while the number of solutions archived in the classical NSGA-II is not limited. As a result, the ACD of the knowledge-informed NSGA-II is very likely to be much smaller than that of the classical NSGA-II. It is unfair to calculate and compare their ACDs directly. To solve this problem, for the Pareto set of the classical NSGA-II, we first rank all solutions based on their crowding distance. The same number of solutions (to the knowledge-informed NSGA-II) with larger crowding distance will be selected. The average crowding distance of these selected solutions is used as the ACD of the classical NSGA-II. The ACDs of the final Pareto set and some intermediate Pareto sets are calculated, respectively.

OFV of each objective: Both ARI and ACD evaluate solution quality when the four objectives are considered together. They cannot reflect how the algorithm works on each objective and show the trade-offs between multiple objectives. Therefore, for each of the final Pareto sets or the selected intermediate Pareto sets, the average OFVs of solutions for the four objectives are calculated and plotted for comparison as shown in Figure 6.

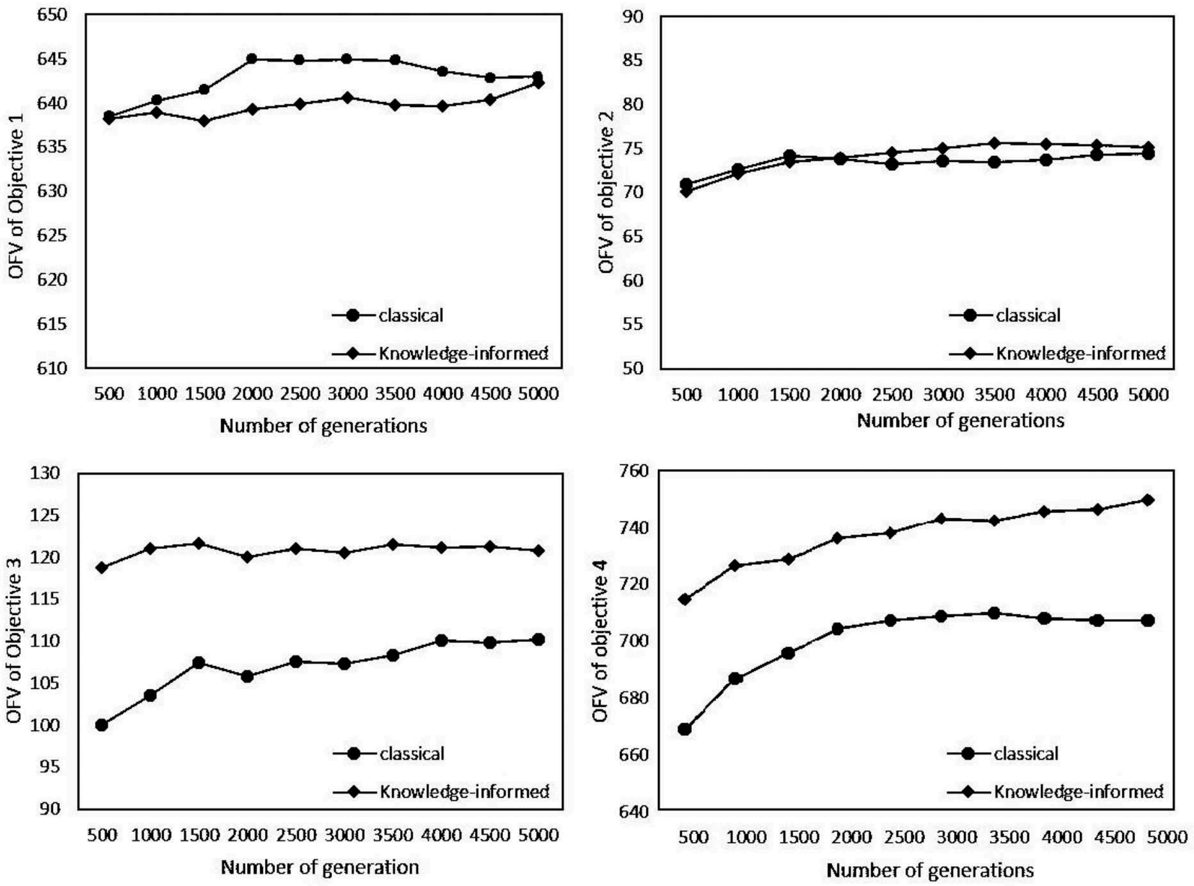

Figure 6. OFV of each objective at different generations.

Mapping pattern of solutions and computation time: except for the above indicators, we also consider the mapping patterns of solutions and the computation time of the two algorithms. To summarize, an effective and efficient algorithm is supposed to produce solutions that are closer to the true Pareto front and have higher diversity within a shorter computation time. Meanwhile, the mapping of solutions should also be reasonable and desirable under the planning context.

2.4.3. *Multiple run performance evaluation*

As heuristic algorithms, both the classical and the knowledge-informed NSGA-II will not produce exactly same solutions in multiple runs. The above measurements compare the performance of the two algorithms based on the solutions they produce in a random run. This is based on the following assumptions: when the two algorithms are compared based on the solutions they produced in different runs, their relative advantage or disadvantage are consistent: one algorithm consistently produces better or worse solutions than the other algorithm in multiple runs.

To test this assumption, we ran each algorithm to solve the same MOLA problem for 15 times. Then, the solutions produced by the two algorithms were coupled to build 15 combinations. Each combination contains two sets of solutions; one set is produced by the classical NSGA-II in one run, and the other set is produced by the knowledge-informed NSGA-II in one run. To keep the diversity of the combinations, for both algorithms, each of the 15 runs was used exactly once. Then, for each combination, the ARIs, ACDs, and average OFVs of the final solutions produced by the two algorithms were calculated and compared.

3. Results and analysis

Both the classical and the knowledge-informed NSGA-II were run to solve the MOLA problem. In both algorithms, the population size was set as 100, and the algorithms were set to terminate at the 5000 generation. The ARIs and ACDs of the final Pareto sets and some intermediate Pareto sets were calculated for the two algorithms, respectively (Table 1). For each algorithm, seven representative solutions were selected from the final Pareto set and mapped for comparison (Figures 7 and 8). The computation time at different generations was recorded and represented in Figure 9.

3.1. *Closeness to the true Pareto front*

For both the final and intermedia Pareto sets, the ARIs of the knowledge-informed NSGA-II all equal to 1, while those of the classical NSGA-II are obviously higher than 1. This indicates that, at different stages of evolution, the knowledge-informed NSGA-II produces solutions much closer to the true Pareto front compared to the classical NSGA-II. At the same generation, the solutions produced by the knowledge-informed NSGA-II are not dominated by any solution produced by the classical NSGA-II.

Table 1. ARI and ACD at different generations.

	ARI		ACD	
Generations	Classical algorithm	Knowledge-informed NSGA-II	Classical algorithm	Knowledge-informed NSGA-II
500	1.79	1	0.081	0.080
1,000	1.72	1	0.081	0.082
1,500	1.74	1	0.081	0.081
2,000	1.48	1	0.082	0.081
2,500	1.64	1	0.082	0.081
3,000	1.65	1	0.082	0.081
3,500	1.61	1	0.082	0.080
4,000	1.58	1	0.081	0.082
4,500	1.58	1	0.081	0.082
5,000	1.61	1	0.081	0.082

3.2. *Solution diversity*

At different stages of the evolution, the ACDs of the two algorithms are similar to each other. This indicates the solution diversities produced by the two algorithms are similar. The knowledge-informed NSGA-II does not sacrifice the solution diversity.

3.3. *OFV of solutions*

As for the solutions in the final Pareto sets (Table 2), the average OFVs of solutions for Objectives 3 and 4 generated by the knowledge-informed NSGA-II are obviously higher than those generated by the classical NSGA-II. The average OFVs of solutions for Objectives 1 and 2 generated by the two algorithms are similar.

At different generations, for Objectives 1 and 2, the average OFVs of solutions generated by the two algorithms are very close. The differences are within 10. As for the solutions generated by the classical NSGA-II, the average OFV for Objective 1 reaches the maximum value at the 2000-generation and then decreases, but the value is still higher than that of the knowledge-informed NSGA-II.

For Objective 3, the average OFV of the solutions generated by the knowledge-informed NSGA-II keeps higher than that of the classical NSGA-II. In the classical NSGA-II, current construction land is not protected from agricultural or construction use. This may result in the lower OFVs for Objective 3.

For Objective 4, the average OFV of solutions generated by the knowledge-informed NSGA-II keeps improving with the increasing generations. In the classical NSGA-II, the average OFV stops to improve after 3500 generations. At the same generation, the average OFV of solutions generated by the knowledge-informed

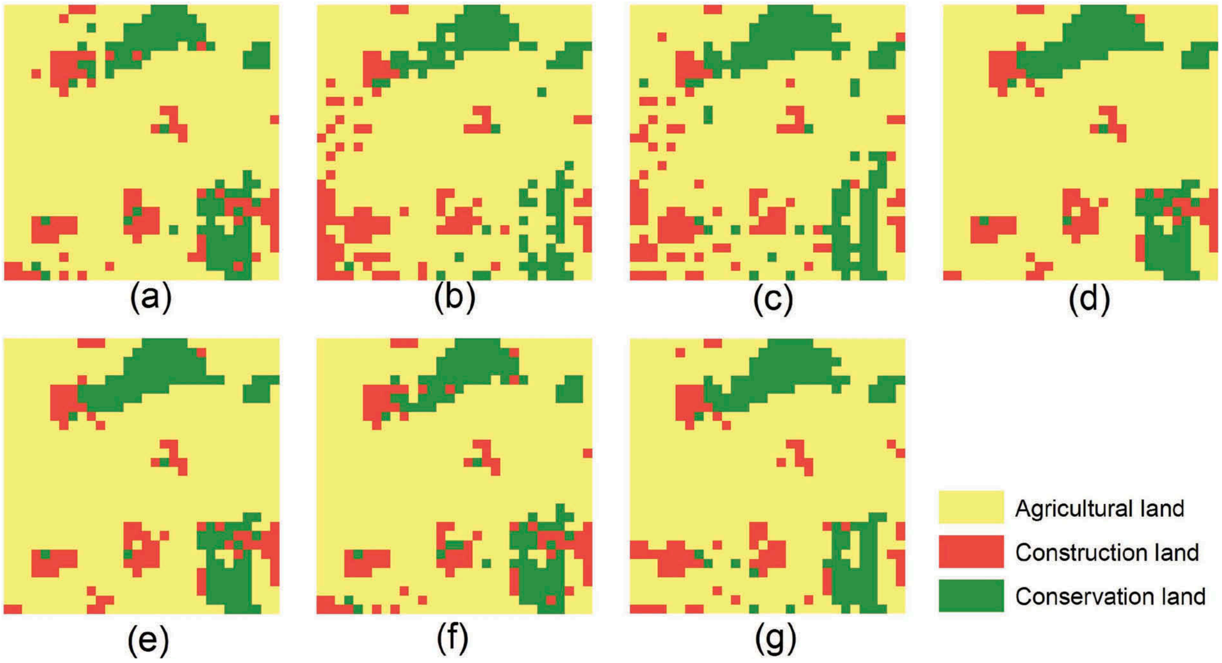

Figure 7. Solutions produced by the classical NSGA-II: (a) agriculture-oriented solution, (b) construction-oriented solution, (c) conservation-oriented solution, (d) solution maximizing spatial compactness, (e) agriculture-oriented solution when spatial compactness is maintained, (f) construction-oriented solution when spatial compactness is maintained, and (g) conservation-oriented solution when spatial compactness is maintained.

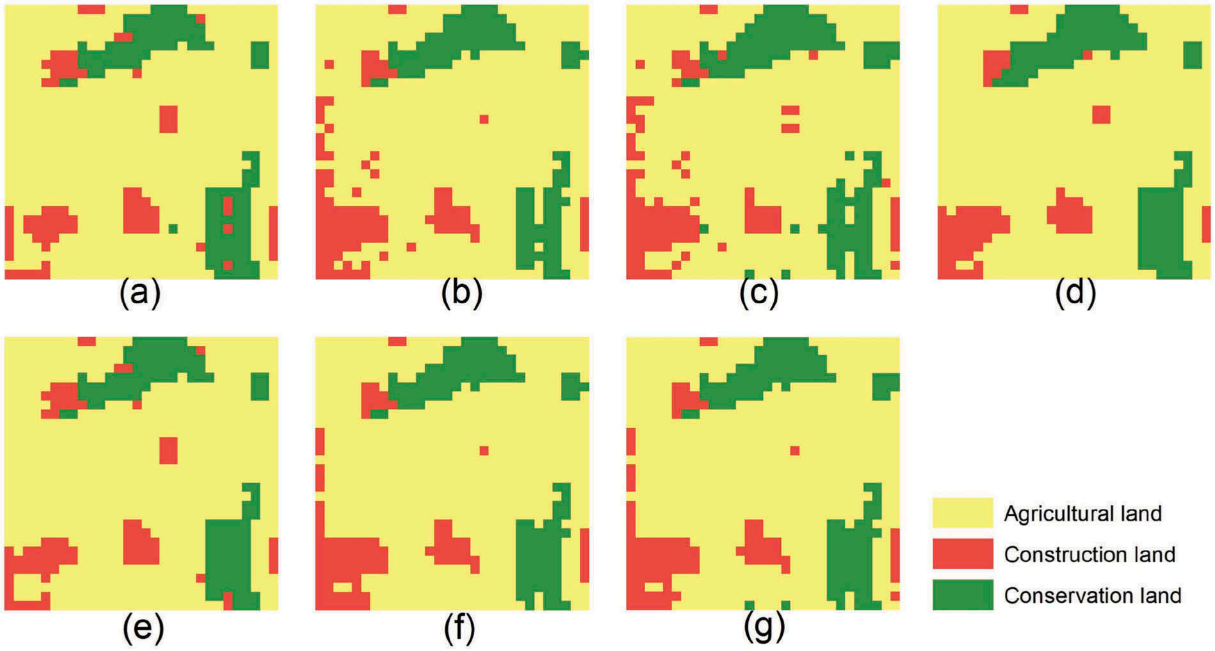

Figure 8. Solutions produced by the knowledge-informed NSGA-II: (a) agriculture-oriented solution, (b) construction-oriented solution, (c) conservation-oriented solution, (d) solution maximizing spatial compactness, (e) agriculture-oriented solution when spatial compactness is maintained, (f) construction-oriented solution when spatial compactness is maintained, and (g) conservation-oriented solution when spatial compactness is maintained.

NSGA-II is obviously higher than that of the classical NSGA-II. The comparison indicates that the knowledge-informed NSGA-II is more effective and efficient in improving the spatial compactness of land allocation.

To summarize, as for Objectives 1 and 2, the knowledge-informed NSGA-II does not show any obvious advantage, but it is obviously more effective and efficient in handling Objectives 3 and 4. The

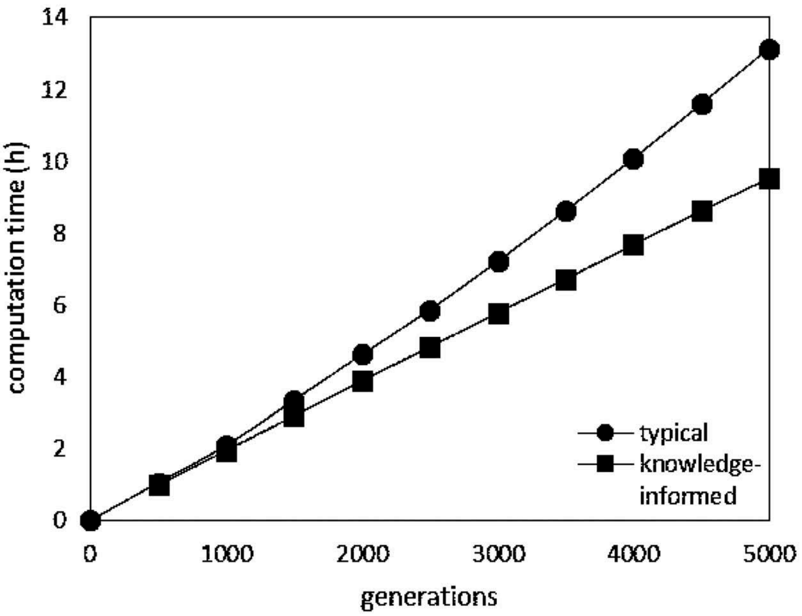

Figure 9. Computation time of classical and knowledge-informed NSGA-II.

Table 2. Final solutions: average OFV of each objective.

NSGA-II Algorithm	Objective 1	Objective 2	Objective 3	Objective 4
classical	642.97	74.42	110.15	707.16
knowledge-informed	642.24	75.13	120.72	749.47

improvement on Objective 4 may sacrifice Objective 1 or 2, but the sacrifice is not too much.

3.4. *Mapping patterns of solutions*

For each algorithm, in the final Pareto set, the four solutions with the highest OFVs on the four objectives are selected and mapped, respectively (Figures 7(a–d) and 8(a–d)). Then we rank all solutions based on their OFVs on Objective 4 and select the top 20 solutions. Among the 20 solutions, three solutions with the highest OFVs on Objectives 1, 2, and 3 are selected and mapped, respectively (Figures 7(e–g) and 8(e–g)). In the planning context, Figures 7(d) and 8(d) are solutions that maximize spatial compactness of land allocation. Figures 7(a,c) and 8(a,c) are agriculture-oriented solutions, Figures 7(b,f) and 8(b,f) are construction-oriented solutions, and Figures 7(c,g) and 8(c,g) are conservation-oriented solutions.

When the solutions from the two algorithms are compared, there are much more scattered or isolated cells in the solutions produced by the classical NSGA-II. This is not desirable in the planning context. In contrast, the land allocation pattern is more compact in the solutions produced by the knowledge-informed NSGA-II.

The further analysis of the solutions produced by the knowledge-informed NSGA-II shows the following results. First, all solutions preserve current conservation land when compared to the current land use map (Figure 1). Second, in the construction-oriented solutions (Figure 8(b,f)), construction land tends to be allocated in the southwest corner of the simulated planning area. This is reasonable because this area is close to the township center. Moreover, when the spatial compactness is maintained, the conservation-oriented (Figure 8(f)) and construction-oriented (Figure 8(g)) solutions look very similar. This indicates that the solutions provide a good compromise between the objectives of maximizing construction suitability and conservation suitability. Finally, the agriculture-oriented (Figure 8(a,e)) and construction-oriented (Figure 8(b,f)) solutions are visually very different, especially in where the new construction land is allocated. This indicates that the objectives of agricultural land protection and construction sprawl are conflicted with each other.

3.5. *Computation time*

Both the classical and the knowledge-informed NSGA-II were programmed with Python 2.7 and run on a computer with Intel Core i7–3770 CPU (3.40GHz). With generation increasing, the computation time of the knowledge-informed NSGA-II grows slower than that of the classical NSGA-II. To finish the 5000-generation evolution, the classical NSGA-II takes 13 h 11 min, while the knowledge-informed NSGA-II takes approximate 9 h.

When this comparison is combined with the analysis in Section 3.1–3.4, we can find that the knowledge-informed NSGA-II obtains better solutions within shorter computation time when compared to the classical NSGA-II.

3.6. *Multiple run performance test*

In all combinations, the ARIs of the knowledge-informed NSGA-II are very close to 1. In most combinations, the ARIs of the classical NSGA-II are obviously higher than that of the knowledge-informed NSGA-II and less close to 1. This indicates that in most combinations, the knowledge-informed NSGA-II produces solutions much closer to the true Pareto front compared to the classical NSGA-II. In all combinations, the ACDs of the two algorithms are very close, all around 0.08. This indicates that the solution diversity produced by two algorithms is very similar. In all combinations, for Objectives 1 and 2, the average OFVs of the classical NSGA-II are slightly higher than that of the knowledge-informed NSGA-II. For Objectives 3 and 4, the average OFVs of the knowledge-informed NSGA-II are obviously higher than that of the classical NSGA-II. These comparison results are consistent with the comparison in Sections 3.1–3.3. Therefore, we can confidently conclude that the comparisons based the solutions produced by the two algorithms in a random run are accountable. Table 3 shows the ARIs, ACDs and average OFVs of the solutions produced by the two algorithms in the 15 combinations.

4. Conclusions

An improved knowledge-informed NSGA-II has been developed to solve the MOLA problem with quality solutions by integrating the patch-based, edge growing/decreasing, neighborhood, and constraint steering rules. The performance of the knowledge-informed NSGA-II is compared to that of the classical NSGA-II through a test with a simulated planning area of 30 × 30 grid.

The knowledge-informed NSGA-II consistently produces solutions much closer to the true Pareto set within a shorter computation time, while the solution diversity is similar to that generated by the classical NSGA-II. When the objectives are individually considered, the knowledge-informed NSGA-II shows an obvious advantage in encouraging compact land allocations.

Based on the solution mapping patterns, the solutions produced by the knowledge-informed NSGA-II are more reasonable and desirable under the planning context: Less scattered/isolated land units are identified, and the solutions provide a good compromise between the objectives of construction sprawl and conservation land protection.

For Objectives 1 and 2, the knowledge-informed NSGA-II does not show obvious advantages, but it is obviously more effective and efficient in handling Objectives 3 and 4. This can be explained by the knowledge-informed designations. First, the crossover and mutation operators incorporate edge-growing/decreasing, patch-based, and neighborhood rules to encourage compact land allocation. The obvious difference on Objective 3 between the solutions produced by the two algorithms reflects the effectiveness of these knowledge-informed designations. Second, in the land suitability data, current conservation land is evaluated highly suitable for conservation. In the knowledge-informed NSGA-II, current conservation land is strictly protected from being converted to other land uses through the initialization and the operators. This results in a higher suitability value in Objective 4. Moreover, since previous research has proven that classical NSGA-II is effective and efficient in handling the non-spatial objective of maximizing land suitability, our knowledge-informed designations do not make any efforts to further improve its performance in maximizing land suitability (Datta et al. 2007; Cao et al. 2011; Song and Chen 2018). This explains why the knowledge-informed NSGA-II does not show any advantages over classical algorithm in improving the OFVs of Objectives 1 and 2. However, OFVs of the two algorithms on Objectives 1 and 2 are very close. This means that the improvement in spatial compactness and conservation suitability does not sacrifice much on agricultural and construction suitability.

The focus of this study to develop an improved knowledge-informed NSGA-II, and compare its performance to that of the classical NSGA-II. If it can effectively and efficiently achieve quality solutions and has obvious advantages over the classical NSGA-II, it will be applied to solve real-life MOLA problems in the future. The simulated planning area represented by the 30 × 30 grid is applied for the convenience of algorithm design and comparison. The improved algorithm has been proven to work effectively and efficiently for the simulated dataset, but there are some challenges when it is applied to real-life planning problems.

First, in the test, the population size was set as 100, and the algorithm was set to terminate at the 5000 generation. Both the population size and the generations of evolution are large enough to make the algorithm converge to quality solutions for the 30 × 30 grid dataset. Meanwhile, the computation time is also acceptable. However, a real-life planning area usually has a larger data size. A smaller population size or a fewer number of generations will decrease the computation time, but they may sacrifice the solution quality. As a result, which population size and generation number can provide a good compromise between the solution quality and

Table 3. Multiple run comparison: ARI, ACD and average OFVs in 15 different combinations.

	ARI		ACD		Average OFV of Objective 1		Average OFV of Objective 2		Average OFV of Objective 3		Average OFV of Objective 4	
Combination	Classical algorithm	Knowledge-informed NSGA-II	Classical algorithm	Knowledge-informed NSGA-II	Classical algorithm	Knowledge-informed NSGA-II	Classical algorithm	Knowledge-informed NSGA-II	Classical algorithm	Knowledge-informed NSGA-II	Classical algorithm	Knowledge-informed NSGA-II
1	1.23	1	0.080	0.081	644.04	638.75	76.37	72.85	113	123	717.78	747.37
2	1.07	1	0.078	0.082	644.74	642.24	76.99	75.13	116	121	728.25	749.47
3	1.56	1	0.082	0.081	646.40	640.97	77.69	72.91	114	122	724.77	745.68
4	1.12	1.06	0.078	0.081	645.99	639.80	77.59	74.13	112	123	723.98	745.57
5	1.19	1.02	0.078	0.082	645.47	641.84	76.81	72.97	115	120	726.31	752.24
6	1.21	1.04	0.081	0.081	646.11	641.80	77.25	73.63	115	121	721.71	750.76
7	1.09	1.01	0.082	0.081	645.07	640.95	76.92	74.29	114	122	718.18	746.13
8	1.3	1.02	0.081	0.082	645.40	641.92	76.58	74.34	112	121	725.85	752.07
9	1.59	1	0.081	0.081	644.72	640.50	77.43	74.90	113	122	720.24	745.13
10	1.58	1	0.081	0.082	644.78	640.03	76.45	74.75	112	122	721.32	747.45
11	1.05	1	0.081	0.082	645.23	641.10	76.68	75.00	112	123	724.49	745.45
12	1.16	1.06	0.081	0.082	645.09	639.40	77.37	72.93	114	121	718.33	751.21
13	1.17	1.03	0.081	0.082	645.47	641.54	76.80	74.48	113	120	722.31	747.13
14	1.09	1.02	0.078	0.082	645.97	639.42	76.38	74.48	113	121	723.13	745.51
15	1.62	1	0.079	0.810	644.07	639.89	76.34	74.01	115	121	727.87	746.03

computation time is one of the challenges when the algorithm is applied to real-life planning problems.

Moreover, each cell in the grid represents a 100 m × 100 m square area in the test. However, when the algorithm is applied to real-life planning problems, the appropriate cell size will depend on the resolutions of available data, the size, physical, and socioeconomic conditions of the planning area, and the requirements of the planning projects. Different planning projects require different spatial resolutions. A higher spatial resolution will result in a larger data size, then require longer computation time. When solving real-life planning problems, the decision makers need to make a compromise between the computation time and the spatial resolutions by considering the planning objectives. This is another challenge.

Besides, different from the square shape of the simulated planning area, the shape of a real-life planning area is usually irregular. The cells on the border of the planning area cannot be directly decided based on their row or column number. These border cells have unique characteristics – some of their neighboring cells are outside the planning area. Therefore, some modifications have to be made to adapt the algorithm for planning areas with an irregular shape, especially in identifying the border cells and applying the neighborhood rules.

Finally, the knowledge-informed NSGA-II is expected to support decision-making in solving MOLA problems, instead of providing final solutions. As a Pareto-based method, it provides solutions that reflect different tradeoffs between multiple objectives, but which solution(s) should be finally selected still requires the knowledge from planners. Moreover, as a heuristic optimization technique, it is not guaranteed to provide optimal solutions. When the algorithm fails to produce satisfactory solutions directly, how to develop knowledge-informed rules/operators to post-process the solutions will also be a challenge. Therefore, our future work will apply the improved knowledge-informed NSGA-II to solve real-life planning problems and try to develop effective and feasible methods to deal with these challenges.

Notes on contributors

Mingjie Song is a Ph.D. candidate in the Department of Geography & Planning, Queen's University, Canada. She received her B.A. and M.S. degrees in geography and land administration from Central China Normal University, China. Her research interests include GIS, land administration, spatial analysis, and spatial optimization.

Dongmei Chen is a professor in the Department of Geography & Planning, Queen's University, Canada. She received her B.A. degree in geography from Peking University, China, and her M.S. degree in GIS and remote sensing application from the Institute of Remote Sensing Application, Chinese Academic of Science. She got her Ph.D. in geography from the joint doctoral program of San Diego State University and University of California at Santa Barbara in 2001. Her research interests include integration of GIS, remote sensing and spatial analysis, land use/cover change detection and modeling, multi-resolution data analysis and classification, spatial dynamic modeling, and spatial-temporal spreading of communicable diseases.

References

Aerts, J. C. J. H., E. Eisinger, G. B. M. Heuvelink, and T. J. Stewart. 2003. "Using Linear Integer Programming for Multi-Site Land-Use Allocation." *Geographical Analysis* 35 (2): 148–169. doi:10.1111/j.1538-4632.2003.tb01106.x

Aerts, J. C. J. H., and G. B. M. Heuvelink. 2002. " Using Simulated Annealing for Resource Allocation." *International Journal of Geographical Information Science* 16 (6): 571–587. doi:10.1080/13658810210138751.

Banks, A., J. Vincent, and C. Anyakoha. 2007. "A Review of Particle Swarm Optimization. Part I: Background and Development." *Natural Computing* 6 (4): 467–484. doi:10.1007/s11047-007-9049-5.

Cao, K., M. Batty, B. Huang, Y. Liu, L. Yu, and J. Chen. 2011. "Spatial Multi-Objective Land Use Optimization: Extensions to the Non-Dominated Sorting Genetic Algorithm-II." *International Journal of Geographical Information Science* 25 (12): 1949–1969. doi:10.1080/13658816.2011.570269.

Cao, K., B. Huang, S. Wang, and H. Lin. 2012. "Sustainable Land Use Optimization Using Boundary-Based Fast Genetic Algorithm." *Computers, Environment and Urban Systems* 36 (3): 257–269. doi:10.1016/j.compenvurbsys.2011.08.001.

Coello, C. A. C., G. B. Lamont, and D. A. Van Veldhuizen. 2007. *Evolutionary Algorithms for Solving Multi-Objective Problems*, Vol. 5. New York: Springer.

Datta, D., K. Deb, C. M. Fonseca, F. G. Lobo, P. A. Condado, and J. Seixas. 2007. "Multi-Objective Evolutionary Algorithm for Land-Use Management Problem." *International Journal of Computational Intelligence Research* 3 (4): 371–384. doi:10.5019/j.ijcir.2007.118.

Deb, K. 2014. *Search Methodolgies: Introductory Tutorials in Optimization and Decision Support Techniques: Multi-Objective Optimization*. New York: Springer. doi:10.1007/978-1-4614-6940-7.

Deb, K., A. Pratap, S. Agarwal, and T. Meyarivan. 2002. "A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II." *IEEE Transactions on Evolutionary Computation* 6 (2): 182–197. doi:10.1109/4235.996017.

Duh, J.-D., and D. G. Brown. 2007. "Knowledge-Informed Pareto Simulated Annealing for Multi-Objective Spatial Allocation." *Computers, Environment and Urban Systems* 31 (3): 253–281. doi:10.1016/j.compenvurbsys.2006.08.002.

Eastman, J. R., H. Jiang, and J. Toledano. 1998. *Multicriteria Analysis for Land-Use Management: Multi-Criteria and Multi-Objective Decision Making for Land Allocation Using GIS*. Dordrecht: Springer.

Knowles, J. D., and D. W. Corne. 2000. "Approximating the Nondominated Front Using the Pareto Archived Evolution Strategy." *Evolutionary Computation* 8 (2): 149–172. doi:10.1162/106365600568167.

Ligmann-Zielinska, A., R. L. Church, and P. Jankowski. 2008. "Spatial Optimization as a Generative Technique for Sustainable Multiobjective Land-Use Allocation." *International Journal of Geographical Information Science* 22 (6): 601–622. doi:10.1080/13658810701587495.

Liu, Y., J. Peng, L. Jiao, and Y. Liu. 2016. "PSOLA: A Heuristic Land-Use Allocation Model Using Patch-

Level Operations and Knowledge-Informed Rules." *PLoS ONE* 11 (6): e0157728. doi:10.1371/journal.pone.0157728.

Masoomi, Z., M. S. Mesgari, and M. Hamrah. 2013. "Allocation of Urban Land Uses by Multi-Objective Particle Swarm Optimization Algorithm." *International Journal of Geographical Information Science* 27 (3): 542–566. doi:10.1080/13658816.2012.698016.

Mohammadi, M., M. Nastaran, and A. Sahebgharani. 2015. "Sustainable Spatial Land Use Optimization through Non-Dominated Sorting Genetic Algorithm-II (NSGA- II): (Case Study: Baboldasht District of Isfahan)." *Indian Journal of Science and Technology* 8 (S3): 118–129. doi:10.17485/ijst/2015/v8iS3/60700.

Sahebgharani, A. 2016. "Multi-Objective Land Use Optimization through Parallel Particle Swarm Algorithm: Case Study Baboldasht District of Isfahan, Iran." *Journal of Urban and Environmental Engineering* 10 (1): 42–49. doi:10.4090/juee.2016.v3n1.042049.

Santé-Riveira, I., R. Crecente-Maseda, and D. Miranda-Barrós. 2008. "GIS-based Planning Support System for Rural Land-Use Allocation." *Computers and Electronics in Agriculture* 63 (2): 257–273. doi:10.1016/j.compag.2008.03.007.

Satty, T. L. 2008. "Relative Measurement and Its Generalization in Decision Making: Why Pair Wise Comparisons are Central in Mathematics for the Measurement of Intangible Factors -The Analytic Hierarchy Process." *RACSAM - Revista De La Real Academia De Ciencias Exactas, Fisicas Y Naturales. Serie A. Matematicas* 102 (2): 251–318.

Shaygan, M., A. Alimohammadi, A. Mansourian, Z. S. Govara, and S. M. Kalami. 2014. "Spatial Multi-Objective Optimization Approach for Land Use Allocation Using NSGA-II." *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 7 (3): 873–883. doi:10.1109/JSTARS.2013.2280697.

Song, M., and D. Chen. 2018. "A Comparison of Three Heuristic Optimization Algorithms for Solving the Multi-Objective Land Allocation (MOLA) Problem." *Annals of GIS* 24 (4): 19–31. doi:10.1080/19475683.2018.1424736.

Tong, D., and A. T. Murray. 2012. "Spatial Optimization in Geography." *Annals of the Association of American Geographers* 102 (6): 1290–1309. doi:10.1080/00045608.2012.685044.