Literature Review of Metaheuristic Applications in Agriculture

Justin Dannemiller and Keith Martin Machina

***Abstract*—Although computer science is a wide-reaching domain that has been applied to a very diverse set of fields, the task of finding an optimal solution is a problem that pervades nearly every domain. There are many situations, however, in which the problem domain is simply too large or too complex for traditional optimization methods. Nature-inspired metaheuristics present a powerful tool for this issue, as they afford intuitive, efficient means by which the optimal solution can be searched for and located. Due to these benefits, such metaheuristics have been applied in a wide range of domains including health care engineering, and agriculture. This paper presents a review of applications of metaheuristics on an important issue in agriculture: land use arrangement. Optimization of land use arrangement considers how the use of lands within an urban region can be arranged such that the constrained amount of land can optimally support a region's population and their requirements. This paper reviews how four metaheuristic algorithms were applied for this purpose.**

***Index Terms*—Metaheuristics, Agriculture, Land use, Multi-objective optimization**

I. Introduction

THE world's human population is growing at an extremely fast pace. Most of this population, moreover, is concentrated in select urban areas. Such urban regions must be able to support the living requirements of their large populations, but, unfortunately, the lands and resources of urban regions in inherently constrained. From a sustainability standpoint, it is therefore imperative the use of urban lands be arranged in an optimal manner to prevent the worsening of humanitarian issues, such as a lack of food, water, housing, and energy.

Optimization of land use arrangements, however, it not a problem that can be solved using only a single factor. There are many parameters that define how optimal a use of a given unit of land is, such as the compactness of land units and the satisfaction of dependencies between neighboring land uses. Due to this, many approaches to this problem have considered multiple objective functions and have optimized the corresponding sets of objective values to locate optimal land use arrangements. This paper reviews four papers which each apply a nature-inspired metaheuristic and section III then concludes with a summary of each of these methods and how they been applied to address the land use problem.

II. Review of Metaheuristic Applications

A. Ant Colony Optimization

Land use change (LUC) is a process of how the use of lands changes over time in a given region. For instance, as a given region develops, unused forested lands often become cleared and utilized for industrial purposes. LUC occurs naturally over time as a region develops and the needs of its comprising population shift. It is important, moreover, to model such change to predict future population requirements. [1] demonstrated a method of modeling the spatial distribution of land use change. Their model was able to predict future distribution of land use and generate a corresponding predictive spatial land use map.

The model was built upon a training set which quantified the spatial distribution of land in a given region at two past years. The model was tasked, moreover with predicted a spatial land use map at a later year using updated spatial attribute data. Three components were integrated to build the model: Ant Colony Optimization (ACO), cellular automata (CA), and a markov chain. A cellular automata is a computational grid of cells where each cell's state changes over time according to a set of local transition rules. These rules inform how the state should change based upon the neighboring cells' states.

Ant Colony Optimization was applied to generate these rules. The goal, moreover, is to produce a set of local transition rules which covers a maximal number of cases from the training set. Each rule is defined as an implication statement in which the antecedent is comprised of multiple conditions. Each ant starts with an empty rule, and probabilistically chooses values for each condition based on cases in the training set. Equations 1 and 2 define this process where $\tau_{\mathbf{ij}}$ and $\eta_{\mathbf{ij}}$ describe the pheromone deposited at value j of condition i at time t and the corresponding quality of that value respectively. The result of these equations is that a condition value that leads to a fewer number of consequences is selected with higher probability.

$$P_{ij} = \frac{\tau_{\mathbf{ij}}(\mathbf{t}) * \ \eta_{\mathbf{ij}}}{\sum_{i=1}^{i=n} \sum_{j=1}^{j=m} \tau_{\mathbf{ij}}(\mathbf{t}) * \ \eta_{\mathbf{ij}}} \tag{1}$$

$$\eta_{\mathbf{ij}} = \frac{maximum \left(freq(T_{ij}^{\mathbf{1}}) \right), \ freq \left(T_{ij}^{\mathbf{2}} \right), \ \ldots freq \left(T_{ij}^{k} \right)}{T_{ij}} \tag{2}$$

Ants continue moving along their rule until all conditions within its antecedent are specified. At this point, the rule consequence is chosen by selecting that which follows the antecedent most frequently in the training set cases. The updating of pheromone concentration at any given condition value is characterized by equations 3 and 4. The Quality term expresses the quality of a given rule and is calculated such that a rule that covers more training set cases has higher quality.

$$\tau_{\mathbf{ij}}(\mathbf{t}) = \tau_{\mathbf{ij}}(\mathbf{t-1}) * (\mathbf{1} - \ \rho) + \left(\mathbf{1} - \frac{\mathbf{1}}{\mathbf{1} + Quality} \right) * \tau_{\mathbf{ij}}(\mathbf{t-1}) \tag{3}$$

$$\mathbf{Q} = \frac{\mathbf{TruePos}}{\mathbf{TruePos} + \mathbf{FalsePos}} + \frac{\mathbf{TrueNeg}}{\mathbf{FalsePos} + \mathbf{TrueNeg}} \tag{4}$$

Next any conditions which contribute neutrally or negatively to the rule are removed, and the resulting set of pruned rules are then added to a local rule database. Then, a Markov chain was created to model the transition process and used to generate a transition area matrix. The resulting transition area matrix describes, for each pair of land uses (A, B), how much land area will transition from land use A to land use B. Next, a cellular automata is created to generate a predictive spatial land use map at a future year where each pixel is considered a cell. The current land use of a given pixel is then mapped to a case in the testing set. The case is then searched for in the transition rule database to find any rules which match the conditions of the case. A probability roulette-wheel is then used to choose any such matching rules, and the consequence of the chosen rule is applied to the pixel to determine its resulting land use. This pixel-remapping process continues until the total degree of change for each type of land use matches that specified by the transition area matrix.

The resulting spatial land use map was compared to the actual land use map of the corresponding year. Their simulated result showed significant resemblance to the actual map with a 72.93% accuracy rate. Their model also out-performed the previous ACO-CA and CA-Markov models.

B. Particle Swarm Optimization

It is of critical importance that land use be optimally arranged in urban regions such that the substantial demands of their population can be met. [2] demonstrated a method by which a set of relatively optimal land use arrangements can be found using multiple objective functions. To guide the search towards optimal arrangements, a modified version of Particle Swarm Optimization was used. The four objective functions that were used are compatibility among neighboring land uses, dependency between neighboring land uses, suitability of a land unit for its assigned land use, and compactness of neighboring land uses. Each of these objective functions were evaluated on the particles independently. The position of each particle, moreover, represents a potential arrangement of lands that could be chosen for the entire region.

Each of the objective functions were independently evaluated on the initial population of particles. The resulting values for each of the four objective functions were then used to develop a 4-dimensional hyperspace. Within this hyperspace, the set of arrangements which are not inferior or dominated by another solution formed the initial Pareto set. PSO was then iteratively applied to guide the particles to more optimal arrangements. Since there was no global best solution, but rather a Pareto set of relatively optimal solutions, the calculation of the global best term was modified as shown in equation 5. The algorithm instead finds a leader, rep(h), to represent the global best. To achieve this, the probability equation 6 is used to choose a hypersphere, h, from the hyperspace. In this equation, m_i describes the number of non-dominated solution in each hypersphere. A hypersphere with less non-dominated solutions within it is chosen with lower probability to prevent early intensification and keep the Pareto set spread out. A random non-dominated solution is chosen from the selected hypersphere to become the leader, rep(h).

$$v_k\left(t\right) = w{*}v_k\left(t-1\right) + c_1{*}r_1\left(P_{best_k} - x_k\right) + c_2{*}r_2{*}(rep\left(h\right)_k - x_k) \tag{5}$$

$$P_i = \frac{e^{-\beta * m_i}}{e^{-\beta * m_1} + e^{-\beta * m_2} + + e^{-\beta * m_H}} \tag{6}$$

The selected leader rep(h), the particle's prior momentum and the particle's best position recorded thus far, are then used to update the velocity of each particle for each objective function using equation 5. The resulting velocity value is then used to update the position of the particle using equation 7. The velocity and position of the particles are repeatedly updated until a maximum number of iterations has elapsed.

$$x_k(t) = x_k(t-1) + v_k(t) \tag{7}$$

The four objective functions used in this optimization problem are compatibility, dependency, suitability, and compactness. Compatibility is the degree to which two neighboring land uses can coexist without interfering with one another. It is calculated using using a compatibility matrix generated from city planner data. The objective function for compatibility is to maximize the sum of total compatibility and the compatibility of the least two compatible neighboring land uses, as shown in 8.

$$f1 : Max\left(\frac{1}{n} * \left(\sum_{i=1}^{i=n} \frac{1}{n_i} * \sum_{j=1}^{j=n_i} \left(Comp_{ij}\right)\right) + Min\left(Comp_{ij}\right)\right) \tag{8}$$

Dependency is the degree to which a land use's dependencies are satisfied by its neighboring land uses. The objective function for dependency is to maximize the sum of total dependency in the arrangement and the dependency score of the lowest-dependency satisfied land-use, as shown in 9.

$$f2 : Max\left(\frac{1}{n} * \left(\sum_{i=1}^{i=n} \frac{1}{n_i} * \sum_{j=1}^{j=n_i} \left(Dep_{ij}\right)\right) + Min\left(Dep_{ij}\right)\right) \tag{9}$$

Suitability measures how well-suited a given unit of land is for a particular use. It is defined using parameters, such as area of land, land pollution, and number of edges. The suitability objective function is to maximize the sum of total suitability and the suitability of the worst-suited land unit in the arrangement, as shown in 10.

$$f3 : Max\left(\frac{1}{n} * \sum_{i=1}^{i=n} S_{i,C_i} + Min(S_{i,C_i})\right) \tag{10}$$

Compactness describes, for each land unit, how many other land units are in its neighborhood that share the same land use. The compactness objective function involves maximizing the total compactness in the arrangement.

Using PSO and these four objective functions, a 4-dimensional hyperspace was created which can be projected onto lower dimensions to obtain Pareto frontiers of relatively optimal land use arrangements. A city planner can find their most optimal solution from the these Pareto frontiers, by specifying weights for each objective function according to the significance they consider for each factor.

C. Knowledge-Informed Pareto Simulated Annealing

Knowledge Informed Pareto simulated annealing [3] is a nature inspired algorithm that borrows ideas from Pareto simulated annealing. Pareto simulated annealing purposes to store the solutions produced that are better than previous solutions and that are not dominated by other solutions. Knowledge based Pareto simulated annealing introduces a concept of knowledge in the whole annealing process. Knowledge about the end goal can be applied or mapped as rules to guide the search towards the global optima and avoid the algorithm to move in a direction that doesn't lead to the global optima. Knowledge based Pareto simulated annealing also applies concepts of pareto front and pareto optimality.The introduced "knowledge" works efficiently and correctly guides the algorithm in the direction of the global optima however it is worth noting that if unnecessary information is provided, it deteriorates the operation of the model completely. The algorithm solves a multi-objective optimization problem which is an NP-Hard problem with the aim of improving computational performance using a metaheuristic. Knowledge based Pareto simulated annealing algorithm is based on equation 11.

$$P(s, sn, T, \Lambda s)\ \alpha\ min(1, e^{X\Omega\omega=1(\lambda s\omega(f\omega(s)-f\omega(sn))/T)}) \tag{11}$$

The formula above expresses the probability of picking a newly generated solution and comparing how good it was compared to the previous one. Also, the algorithm purposes to determine if the solution has been dominated or not by other solutions. The specifics of the algorithm are as follows: $P(s, sn, T, \Lambda s)$ represents the probability of accepting a new solutionfrom current to next state. T represents the temperature. Term inside the exponential measures the difference between the objective function of the current state fw(s) and the proposed state fw(s_n) weighted by Λ. $X\ \Omega\ \omega = 1$ represents the sum of the differences all over the objective function . The algorithm was tested on a map generation problem. The task was to generate a map pattern on an 18 X 18 cell landscape given two cover classes, dark and grey such that the two cover equal portions of the landscape. Based on this task, some auxiliary knowledge was introduced to guide the map generation. The knowledge was informed by two rules namely: compactness and contiguity. Compactness moves randomly selected cells to a location promoting compactness of the target cover. That is a location that has the highest number of neighbouring cells with the same cover type Contiguity moves the cells randomly to a location that promotes patch connectivity of the target cover type. The two rules inform knowledge and guide the algorithm into accurately generating the map. It is clearly evident the knowledge was helpful. However, if valueless information is introduced, the algorithm would have a hard time getting to the desired goal. The results were as follows.

The portion of the map on the left was generated using Knowledge based Pareto simulated annealing algorithm while the one on the right was generated through manual editing. The portion of the map on the right generated through manual editing did not meet the requirements.

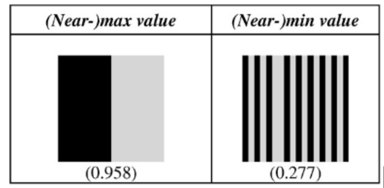

Fig. 1: Map pattern generation using auxiliary knowledge. [3]

D. Boundary-based Fast Genetic Algorithm

Boundary-based Fast Genetic Algorithm [4] a special type of genetic Algorithm that uses weighted sum to construct a Multi-objective optimization land use model as it seeks to assign various weights of various parameters affecting the MOLU It is used to solve the land use allocation problem quite efficiently. It also takes into account the Pareto concept which ensures optimal solutions are not dominated by other solutions When implementing this algorithm First, the objective and constraint should be provided as this guides the algorithm. Planning can be represented in an n by m grid. K can be used to represent the different types of land uses that exist assigned to a cell(i,j) presented as binary. It is used to show that land use has already been assigned on that cell hence 0 shows otherwise. B ijk is set as the parameter of different objectives and it varies with location as it depends on specific attributes of the area according to each objective.

$$F_{obj} = \sum_{i=1}^{N}\sum_{j=1}^{M}\sum_{k=1}^{P} A_0 * B_{ijl} * X_{ijk} \tag{12}$$

III. Conclusion

In conclusion, the use of nature-inspired heuristics has proven to be a promising approach in solving problems in various domains, agriculture being one of them. The focus of this report was on the application of metaheuristics in agriculture using four main algorithms namely: Particle swarm optimization, Ant Colony Optimization, knowledge informed pareto heuristics and Boundary-based Fast Genetic Algorithm. These algorithms were used to mainly solve land allocation problems, and they all demonstrated the efficiency and effectiveness in optimization.

References

[1] X. Yang, X.-Q. Zheng, and L.-N. Lv, "A spatiotemporal model of land use change based on ant colony optimization, markov chain and cellular automata," *Ecological Modelling*, vol. 233, pp. 11–19, 2012. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0304380012001184

[2] Z. Masoomi, M. S. Mesgari, and M. Hamrah, "Allocation of urban land uses by multi-objective particle swarm optimization algorithm," *International Journal of Geographical Information Science*, vol. 27, no. 3, pp. 542–566, 2013. [Online]. Available: https://doi.org/10.1080/13658816.2012.698016

[3] J.-D. Duh and D. G. Brown, "Knowledge-informed pareto simulated annealing for multi-objective spatial allocation," *Computers, Environment and Urban Systems*, vol. 31, no. 3, pp. 253–281, 2007, geoComputation 2005. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0198971506000676

[4] K. Cao, B. Huang, S. Wang, and H. Lin, "Sustainable land use optimization using boundary-based fast genetic algorithm," *Computers, Environment and Urban Systems*, vol. 36, no. 3, pp. 257–269, 2012. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S019897151100086X