2014 IEEE Congress on Evolutionary Computation (CEC)
July 6-11, 2014, Beijing, China

A Cultural Algorithm for Spatial Forest Harvest Scheduling

Wan-Yu Liu and Chun-Cheng Lin

***Abstract*— This paper proposes a cultural algorithm for the spatial forest harvest scheduling for maximizing the total harvested timber volume, under the constraints of minimum harvest age, minimum adjacency green-up age, and approximately even volume flow for each period of the schedule. In order to increase the solution-search ability, the cultural algorithm extracts problem-specific information during the evolutionary solution search to update the belief space of a generation, which has cultural influences and guidance on the next generation. The key design of our cultural algorithm is to propose the cultural and evolutionary operators specifically for the problem. Experimental analysis shows that our cultural algorithm performs better than the previous approaches.**

I. Introduction

FORESTS provide multiple functions–production, protection as well as recreation, so that forest harvest scheduling has been attracting a lot of attention. Forest harvest scheduling becomes more complicated as multiple economic, environmental, and social criteria are taken into account. Among those criteria, the spatial concern in forest harvest scheduling is of importance as it maintains a number of environmental and ecological conditions, such as maintenance of biodiversity, limited sediment loading in streams, limited disruption of habitats in an area, limited impact on a viewshed, supply of open forage areas for certain animals, and so on [1]. Furthermore, various types of damages or spatially uncontrolled management implementations can result in decreased wood quality, habitat disruption, water pollution, increased sediment quantities, and so on. Based on the above reasons, it is common that the spatial constraints on minimum adjacency green-up age are imposed upon harvesting activities on adjacent forest stands.

The focus of this paper is on a spatial forest harvest scheduling problem which aims at maximizing the timber volume harvested over a harvest planning schedule with the consideration of the minimum harvest age constraint, the minimum adjacency green-up age constraint, as well as the constraint of approximately even volume flow for each period of the schedule. In the previous literature, several solutions have been used for solving different types of spatial forest harvest scheduling problems, among which exact solutions include metroplis algorithm [2], mixed integer programming [3], [4], and dynamic programming [5]; while metaheuristic approaches include penalty function with simulated annealing [6], tabu search [7], and evolutionary program [8], among others.

Wan-Yu Liu is with the Department of Tourism Information, Aletheia University, New Taipei City 251, Taiwan (email: nellyliu@gmail.com).
Chun-Cheng Lin is with the Department of Industrial Engineering and Management, National Chiao Tung University, Hsinchu 300, Taiwan (email: cclin321@nctu.edu.tw).

In this paper, a cultural algorithm is proposed to solve the above spatial forest harvest scheduling problem. The cultural algorithm [9] is an evolutionary program [10], [11] which improves the performance of evolutionary search by extracting the domain knowledge of the concerned problem during the search process. Besides the conventional evolutionary settings, it maintains a belief space consisting of a half of individuals with better fitness values from the current generation, as well as a leader to guide the whole population. During the search process, the belief space is updated by incorporation of the extracted problem-specific knowledge, and it influences each individual in the population to obtain better solutions. As a result, this paper investigates how to design a cultural algorithm specifically for the concerned spatial forest harvest scheduling problem with the above concerns. For performance evaluation, the proposed algorithm is experimentally compared with the previous best-known simulated annealing approach to the same problem in [8]. Our experimental results show that our proposed algorithm performs better than the existing approaches.

II. Problem Setting

This section gives the basic settings of our concerned spatial forest harvest scheduling problem. Our concerned spatial forest harvest scheduling problem is the same with [8], which is described as follows. Consider a forest land that consists of a number of smaller polygonal forest lands, called *polygons*, in which any two neighboring polygons are said to have an adjacency relation. For simplicity of the problem, it is supposed that the forests in each polygon are at an equal age, and harvests occur at the beginning of a planning period. We serve as the role of the forest planner who aims at planning a harvest schedule of the forest land that is divided into a number of time periods.

With the above setting, our concerned problem is to select a number of forest polygons to be harvested in the beginning of each period, such that the total volume harvested over the planning harvest schedule is maximized, subject to minimum harvest age constraint; minimum adjacency green-up age constraint; the constraint of approximately even volume flow for each period of the harvest schedule.

The details of the three constraints are stated as follows. The minimum harvest age constraint allows the harvest of only the polygons at age greater than a minimum age threshold. The minimum adjacency green-up age constraint considers the adjacency rule because the harvest should be dispersed for hydrological and wildlife reasons about concentrated harvests associated with progressive clearcutting. The green-up age is the age that a regenerated stand must reach before an adjacent unit can be harvested. Our concerned problem is subject to a minimum adjacency green-up age constraint, in which a forest polygon can be harvested only when the age of each of its adjacent forest polygons is no less than the minimum green-up age. In order to balance the harvest volume of each period, the even flow constraint enforces the timber volume for each period to be harvested as even as possible.

Since forest polygons are harvested only in the beginning of each period, the minimum adjacency green-up age constraint is always satisfied if we let the length of each period be greater than the minimum adjacency green-up age. In this paper, we continue using the setting in [8], in which the minimum harvest age is 90 years, and the length of each period is 20 years, which are always greater than the minimum adjacency green-up age–15 years. With this setting, it suffices to consider the adjacency relationship of polygons in solving the problem, i.e., we only need to consider that any two adjacent polygons cannot be harvested in the same period, with no need to check their age difference.

III. Our Approach

Our approach to the forest harvest scheduling problem is based on the cultural algorithm (CA) [9], which is a class of evolutionary program based on some theories from sociology and archaeology that try to formulate cultural evolution. The CA is carried out with two spaces: *population space* is a set of individuals, in which each individual has a set of independent features used for calculating its fitness, while *belief space* stores the knowledge acquired by individuals through generations. In each iteration of the algorithm, the individuals in the population space can be replaced by some of their descendants, who are obtained by applying some operators to the population and may be influenced or guided by the belief space. Hence, the belief space for each population should be updated by communication with the population.

The basic concept behind our CA approach works on a population space where the best individual is selected as the leader of the belief space and a group of better individuals are selected as the normative matrix of the belief space. The belief space has situational and normative influences on the population space. In addition, the population space is adjusted by both the conventional evolutionary operations (including selection, crossover, repairing, and balancing operations) and the cultural exploration operations (interchange, sequencing, and simple mutation).

Remind that a CA works on a population space and a belief space. In our CA, the belief space needs more delicate designs, while the population space is based upon the EP for the same problem proposed in [8], whose solution representation is explained as follows. Consider a forest land consisting of n forest polygons throughout the rest of this paper, which are labeled by $1, 2, ..., n$. Any feasible solution of the planning problem is represented by an individual's chromosome that is a permutation $\langle x_1 x_2 ... x_n \rangle$ of n genes (each of which represents a polygon ID), where $\forall i, x_i \in \{1, 2, ..., n\}$, and those genes are classified into m period partitions plus a residual partition, as shown in Figure 1. For $1 \leq i \leq m$, the polygons in the i-th period partition are harvested in the beginning of the i-th period of the harvest schedule, while the polygons in the residual partition are never harvested.

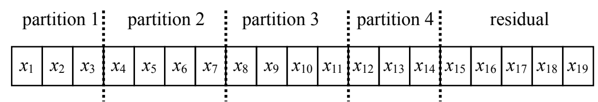

Fig. 1. An example for solution representation.

The main steps of our CA are stated in Algorithm 1, which takes into account *situational influence*, *normative influence*, as well as three cultural exploration operators (*interchange*, *sequencing*, and *simple mutation*). Since the population space of our algorithm is based upon the EP in [8], there exist some common ingredients between the two algorithms. The rest of this section only states in detail the other steps of our CA that are different from [8]: initialization and update of the belief space (Lines 3 and 21, respectively), mutation operators with situational and normative influences (Lines 7 and 9, respectively), and exploration operators (Line 10–17).

Algorithm 1 Our Cultural Algorithm()

1: generate random individuals (schedules) in the initial population
2: evaluate the fitness of each individual in the initial population
3: initialize the belief space (i.e., copy the best individual to the situational belief space, and create the normative matrix from the individuals with fitness greater than the average fitness)
4: **repeat**
5: apply selection operator
6: apply crossover and repairing operators
7: calculate the cumulative probability distribution for roulette-wheel normative influence
8: **for** each individual in current population **do**
9: apply cultural mutation operator (by situational and normative influence)
10: **switch** (exploration operator)
11: **case** interchange:
12: apply interchange operator
13: **case** sequencing:
14: apply sequencing operator
15: **case** simpleMutation:
16: apply simple mutation operator
17: **end switch**
18: **end for**
19: apply balancing operator
20: check if each individual is accepted according to acceptance criteria
21: update the belief space (with the current individuals)
22: **until** the end condition or the maximum iteration is not reached

A. Initialization and Update of the Belief Space

Let η denote the number of individuals in a population. The belief space contains $\eta/2$ individuals that are copied from those individuals in the population with fitness greater than the average fitness of the current population. In addition, the *leader* of the belief space is defined as the individual with the best fitness in the space. Hence, in both the initial setting and each iteration of the algorithm, the belief space and the leader should be updated (Line 21 of Algorithm 1), and they will be used in the mutation operators with normative and situational influences, respectively.

B. Mutation operators with cultural influence

The mutation operators with cultural influence randomly selects a gene position for each individual and mutates it with the situational influence and normative influence, which are stated in detail in the following subsections.

1) Situational influence: The basic idea of the cultural evolution with the situational influence is originated from the phenomenon that each individual in a generation tries to behave as a leader. Following such an idea, our mutation operator with situational influence is designed as follows. For each individual, the operator randomly selects a period partition i of the leader in the belief space and randomly selects a polygon x_j in the partition. If polygon x_j is not located in partition i of this individual, remove polygon x_j in its original partition and then add it to partition i. By doing so, the selected polygon in the individual is located in the same partition as the leader. Note that the generated individual may violate the minimum adjacency green-up age constraint. Hence, before executing the move due to situational influence, we have to check if the violation happens. If the adjacency rule is violated, then the move is canceled.

2) Normative influence: The basic idea of the cultural evolution with normative influence is originated from the phenomenon that a group of individuals selected from the population have an influence on the normative behaviors of the whole population. In our CA, the selected group include the individuals with fitness greater than the average fitness, and they are saved in the belief space, which is updated in each iteration (Line 21 of Algorithm 1).

We use roulette-wheel rule for the mutation operator with normative influence. Since the belief space is corresponding to a roulette wheel, all the individuals in a generation (iteration) use the same roulette wheel, i.e., the wheel is updated once (Line 7 of Algorithm 1) and can be used for all the individuals (Line 9 of Algorithm 1) in each iteration.

In Line 7 of Algorithm 1, the roulette wheel is established as follows. We record the gene in each partition that has the maximal frequency among all the individuals in the belief space. For $i \in \{1, ..., m\}$, let g_i denote the gene in partition i with the maximal frequency $f(g_i)$ for all the individuals in the belief space. Then, the ratio of g_i in the roulette wheel is $f(g_i)/\sum_{j=1}^{m} f(g_j)$.

In Line 9 of Algorithm 1, a random real number between 0 and 1 is selected for each individual. Then we find the region in the roulette wheel where the random real number is located. Without loss of generality, assume that the individual selects the region representing gene g_x. Then, the individual adds gene g_x in partition x.

Note that, different from the situational influence, the operator of normative influence may violate both the minimum harvest age constraint and the adjacency rule. Hence, the two constraints should be checked before each move. If there is any violation, the move of the gene is canceled.

C. Exploration operators

We use the exploration operators for maintaining the diversity of the population. By analogy with [12], we design our three exploration operators as follows.

The first exploration operator is the *sequencing operator*, which arbitrarily finds two neighboring period partitions, and then swaps all the genes of the two partitions. It has an advantage that the generated individual only needs to be repaired by considering whether the moves of the genes from the former partition to the latter partition satisfy the minimum harvest age constraint, because the genes in the same partition satisfy the minimum adjacency green-up age constraint no matter where the partition moves, and the moves of the genes from the latter partition to the former partition still satisfy the minimum harvest age constraint. Note that this operator is the most destructive among the three exploration operators. The second exploration operator is the *interchange operator*, which interchanges two genes respectively from two different period partitions. The third exploration operator is the *simple mutation operator*, which moves a genes from a period partition to another period partition. Note that both the minimum harvest age and minimum adjacency green-up age constraints should be checked for the two operators. If any constraint is violated, the interchange is not executed. By analogy from [12], a parameter control mechanism is applied for selecting the above three exploration operators.

IV. Implementation and Experimental Results

This section addresses the used simulation environment and the implementation of our approach, and then conducts a detailed experimental analysis.

A. Implementation and Simulation Environment

In order to evaluate the performance of our proposed CA approach, we implemented not only our proposed CA approach but also the previous SA approach [8], which has been known to be the best approach to the proposed problem so far. The parameter settings used in our CA approaches and our problem are given as follows: population size: 10, 20, 40 (default: 20); crossover rate: 0.25; number of maximal iterations: 10000, 1500, 2000 (default: 1000); number of runs: 20, 100 (default: 20); minimum harvest age threshold: 90 years; minimum adjacency green-up age: 15 years; length of each period: 20 years.

We conducted all the experiments on an artificial problem instance, which is generated in the following way. At first, we create a 20×20 grid graph, in which each vertex represents a forest polygon; each edge represents that the two forest polygons corresponding to the two end vertices of the edge are adjacent geographically. Note that the degree of each vertex in the grid graph is four. Next, we randomly remove 20 vertices from the graph. Finally, we randomly shrink 80 edges, and obtain the final testing graph. By doing so, the final graph is capable of demonstrating some degree of randomness while maintaining some degree of regularity, such that the characteristics of adjacent forest polygons can be established. In addition, some vertices in the final graph are not necessarily of degree four, e.g., the vertex degrees of the testing graph include one to seven, which increase the diversity of adjacency relationship.

In addition to the adjacency relationship of forest polygons, our problem instance requires the information of area sizes and forest ages. Hence, in our setting for the problem instance, each vertex (representing a forest polygon) is associated with a random real number from $\mathcal{U}(16, 20)$ for area size (ha.) and a random integer from $\{0, 1, \cdots, 99\}$ for forest age (years), where $\mathcal{U}(a, b)$ is the uniform probability distribution function over the range $[a, b]$. For example, if there is a vertex associated with 16.77 ha. and 67 years, it means that the area of the polygon corresponding to this vertex is 16.77 ha., and all the forests planted on this polygon are 67 years old.

The objective of our concerned problem is to maximize the total harvested timber volume, and hence, we let the fitness in our CA approach be the total harvested timber volume. From [13], each hectare of Cunninghamia lanceolata at age t in Taiwan can harvest the following timber volume (m^3): $578.6851 \cdot (1 - t^{-1.5402})^{54.3344}$ where t is the age of Cunninghamia lanceolata.

B. Experimental Results and Discussion

A variety of experiments for comparing the performance of SA and CA and their detailed statistics are given in Table I, in which the SA is experimented with 1000, 1500, 2000 iterations, while our CA is experimented with all the possible combinations of the three maximal iterations and three population sizes (10, 20, 40) as listed in the second and third columns. Note that each entry in Table I is computed by averaging the results of executing 20 times of the considered algorithm. We observe from the 'Best fitness' column (solution) that the CA always performs better than the SA, no matter how many iterations are applied. We observe from the 'Difference ratio' column that more iterations do not imply better performance, in which the cases of the CA with 1500 iterations are the best.

V. Conclusions

A CA-based approach to the spatial forest harvest scheduling problem has been proposed to maximize the total harvested forest timber volume over a planning harvest schedule while some constraints are satisfied. The main design of the algorithm is to add a belief space to the evolutionary program, in which a leader and a normative matrix guide or have situational and normative influences on the whole population to find better solutions. By experimental simulation, it is showed that our proposed cultural algorithm performs better than the simulated annealing approach which was the previous best-known approach to the concerned problem.

TABLE I

Statistics of experimental results for SA and CA.

	Max iteration	Pop. Size	Best fitness	Average fitness	Worst fitness	StdDev. of best fitness	Run. time (s)	Diff. ratio*
SA	1000	–	2519015.00	2460310.14	2424227.00	23947.17	0.33	0.00%
CA	1000	10	2535645.25	2498968.30	2463530.25	13927.56	4.34	0.66%
		20	2536393.25	2507089.96	2477722.00	15622.12	8.13	0.69%
		40	2536046.75	2511164.91	2495069.50	11468.91	17.28	0.68%
SA	1500	–	2498293.75	2461271.63	2395781.00	26761.83	0.49	0.00%
CA	1500	10	2536715.25	2502601.44	2468234.75	18039.91	6.42	1.54%
		20	2547641.25	2507295.95	2486404.25	13776.92	11.83	1.98%
		40	2536502.25	2515781.78	2502841.25	9311.72	24.76	1.53%
SA	2000	–	2509398.25	2447236.36	2374973.50	31961.42	0.67	0.00%
CA	2000	10	2530186.75	2499073.99	2464081.50	15663.53	8.40	0.83%
		20	2530089.00	2511227.08	2491883.50	9894.33	15.37	0.82%
		40	2534226.00	2514793.36	2489267.75	11475.18	32.16	0.99%

* The difference ratio is the measure of the difference of the best fitness from the SA's best fitness over the SA's best fitness.

Acknowledgements

This work has been supported in part by NSC 102-2410-H156-008 and NSC 101-2628-E-009-025-MY3, Taiwan.

References

[1] R. Church, A. Murray, and A. Weintraub, "Locational issues in forest management," *Location Science*, vol. 6, pp. 137–153, 1998.

[2] P. Van Deusen, "Multiple solution harvest scheduling," *Silva Fennica*, vol. 33, no. 3, pp. 207–216, 1999.

[3] B. Bare and L. Eldon, "An evaluation of integer programming in forest production scheduling problems," *Research Bulletin, No. 847, Purdue University Agricultural Experiment Station, Lafayette, IN*, p. 7 p., 1969.

[4] M. McDill and S. Rebain, "A mixed-integer formulation of the minimum patch size problem," *Forest Science*, vol. 49, no. 4, pp. 608–618, 2003.

[5] H. Hoganson and J. Borges, "Using dynamic programming and overlapping subproblems to address adjacency in large harvest scheduling problems," *Forest Science*, vol. 44, no. 4, pp. 526–538, 1998.

[6] C. Lockwood and T. Moore, "Harvest scheduling with adjacency constraints: A simulated annealing approach," *Canadian Journal of Forest Research*, vol. 23, pp. 468–478, 1993.

[7] A. Murray and R. Church, "Analyzing cliques for imposing adjacency restrictions in forest models," *Forest Science*, vol. 42, no. 2, pp. 166–175, 1999.

[8] G. Liu, S. Han, X. Zhao, J. Nelson, H. Wang, and W. Wang, "Optimisation algorithms for spatially constrained forest planning," *Ecological Modeling*, vol. 194, pp. 421–428, 2006.

[9] R. Reynolds, "An introduction to cultural algorithms," in *Proceedings of the 3rd Annual Conference on Evolutionary Programming*, 1994, pp. 131–139.

[10] D. Fogel, *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence*. The Institute of Electrical and Electronic Engineers, New York, 1995.

[11] X. Yu and M. Gen, *Introduction to Evolutionary Algorithms*. Springer, 2010.

[12] C. Soza, R. Becerra, M. Riff, and C. Coello, "Solving timetabling problems using a cultural algorithm," *Applied Soft Computing*, vol. 11, no. 1, pp. 337–344, 2011.

[13] L.-C. Chen and G.-M. Hwang, "Evaluating timber value of China-fir plantations at Lien-hua-chih with weibull probability density function," *Bulletin of Taiwan Forestry Research Institute*, vol. 7, no. 3, pp. 221–230, 1992.