INSTITUTO
DE INFORMÁTICA
UFRGS

A GENETIC ALGORITHM FOR FAIR LAND ALLOCATION

Alex Gliesch, Marcus Ritt, Mayron Moreira

GECCO 2017 — July 2017

AGENDA

1. Introduction
2. Proposed genetic algorithm
3. Computational experiments
4. Conclusions

Introduction

- Agrarian reform (in Brazil): redistribution of land from large latifundia to small, family farmers
- A difficult problem: ethical and legal considerations must be considered
- Subdivisions must respect natural reserves, and lots must be balanced w.r.t. area, soil quality and access to rivers
- Successful implementations of agrarian reform are known to have direct impacts on land use, food production, job creation and eradication of hunger
- Current methods for land subdivision are manual, time-consuming and often flawed

- **Input**
  - A matrix of spatial units $U = [n] \times [m]$.
  - $U = L \cup R \cup P$ (land, rivers and natural reserves)
  - Each $u \in L$ has a soil productivity q_u.
  - The number of desired lots k.
- **Solution:** a k-partition $L = \bigcup_{i \in [k]} C_i$ satisfying:
  - *Connectivity:* each C_i is connected (4-neighborhood in L).
  - *Accessibility:* no enclaves.
  - *Equality:* $\lambda \leq \bar{\lambda} = 3$; λ: ratio of largest to smallest lot area.
  - *Balance:* lots with access to rivers are smaller than lots without.
- **Objective:** minimize std. dev. of lot qualities v_i, where

$$v_i = \sum_{u \in C_i} q_u$$

Proposed genetic algorithm

SOLUTION REPRESENTATION

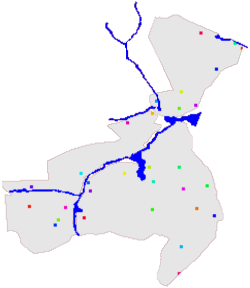
- Solution: assignment of units to lots
- Additionally: data structures for manipulating and visiting border units in constant time
- We also maintain:
  - Total area excess A of lots next to a river which exceed the area of the smallest lot without access
  - Equality excess $\lambda^+ = \max(\lambda - \bar{\lambda}, 0)$
  - Objective value $SS = \sum_{i \in [k]} (v_i - \bar{v})^2$
- Objective function: $\varphi = (A, \lambda^+, SS)$ in lexicographic order

SELECTION STRATEGY

- Generational selection strategy, with a population of size P composed of:
  - eP best individuals of the current generation
  - oP individuals generated by recombination and mutation
  - rP random individuals
  - $e + g + r = 1$
- Parents for recombination are selected by 3-tournament

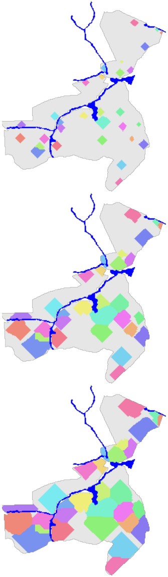
1ST PHASE: SELECTING INITIAL SEEDS

- A greedy constructive algorithm to generate initial solutions
- Two phases:
  1. Select k units as initial seeds from which to grow lots
  2. Grow lots until all reachable free space is assigned



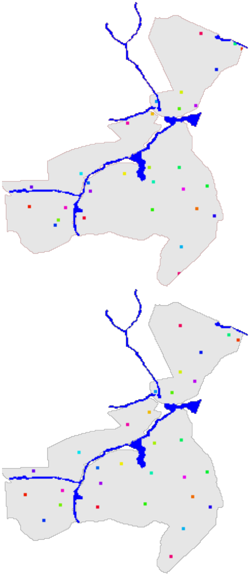
- Land may have multiple connected components, separated by rivers and natural reserves
- Each a lot cannot be in more than one component
- The number of seeds is proportional to the area of the component

1ST PHASE: SELECTING INITIAL SEEDS



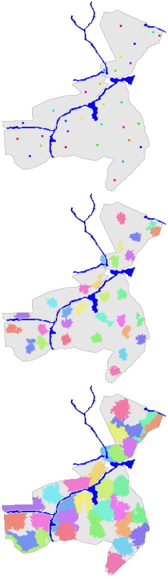
Grow seeds with a shortest distance search on the graph induced by a 4-neighborhood in L, where an edge (u, u') has cost $q_{u'}$:

- Starting from the initial seeds, repeatedly assign the free border unit with smallest q_u to a neighboring lot.
- Ties in neighboring lots are broken by order of neighbor insertion
- Stop when there are no free units

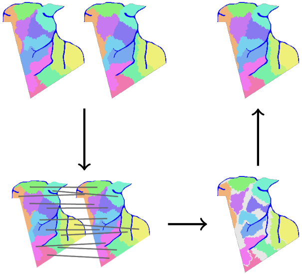


- Recompute each seed as the unit in L closest to the weighted centroid (w.r.t. q) of the grown lot
- This process can be repeated more times (like k-means), but we found that it did not make a significant difference

2ND PHASE: GROW LOTS GREEDILY

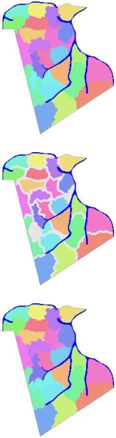


- Starting from the k seeds, repeatedly perform the best feasible assignment w.r.t. the objective φ
  - Assigning a unit to lot i is feasible if C_i remains connected and accessible
  - Connectivity is always ensured, by only considering units that border lots
- Main bottleneck: recomputing φ for each candidate, even when its value changes very little between steps
- $\Rightarrow$ For better performance, we assign batches of b units at every construction step



- Idea: establish best correspondence between lots in p_1 and p_2
- Build a bipartite graph that connects every lot i in p_1 to every lot j in p_2 with weight $|C_i^{p1} \cap C_j^{p2}|$, and find maximum weight perfect matching
- Rebuild full solution using the constructive algorithm starting from the intersection of matched lots

MUTATION



- Remove all units within a given distance from the borders between lots (2 spatial units)
- Reassign borders with greedy construction seeded with the remaining "core lots"
- Special case: if removing the borders splits a lot into multiple components, we keep only the largest one
- Introduce variability in the construction by selecting b units with uniform probability among the best αb candidates (instead of simply the b best)

Computational experiments

- 5 real-world instances
- 25 artificial instances, with 5 different topologies:
  - $n \times n$, with $n \in \{200, 300, 400, 500, 600\}$
  - $k \in \{20, 40, 60, 80, 100\}$
- 4 experiments:
  1. Find optimal batch size b for the constructive algorithm
  2. Scalability: evaluate performance of GA for different problem sizes
  3. Effectiveness: compare the constructive algorithm, the GA, and a simple algorithm based on expansion by BFS
  4. Comparison to manual allocation
- 30 min. running time and five replications for each test

- Parameters calibrated using the irace package in GNU R, with 1000 runs and 5 minutes per run:

Description	Initial range	Best value
Population size P	$[10, 50]$	15
Mutation α	$[2, 5]$	3.72
Elite rate e	$[0, 1]$	0.59
Offspring rate o	$[0, 1]$	0.38
Random rate r	$[0, 1]$	0.03

Batch	Fixed time				50 replications			
size	$A\ (m)$	λ	σ	Repl.	$A\ (m)$	λ	σ	t (s)
32	40.3	4.3	44.6	2,494	28.7	4.0	35.7	668
64	28.4	3.8	27.6	6,118	35.5	3.8	29.3	301
128	28.4	3.7	32.0	13,760	34.4	4.3	36.7	120
256	23.3	4.1	31.8	27,102	44.2	5.5	42.0	52
512	28.3	4.3	44.5	45,790	53.8	4.0	34.7	26
1,024	42.2	4.2	41.5	67,831	73.6	4.8	36.8	15
2,048	51.1	4.5	65.4	88,424	94.3	6.8	58.8	10
4,096	67.9	6.4	82.4	111,203	118.7	6.3	81.6	8

EXPERIMENT 1: OPTIMAL BATCH SIZE

- Fixed number of replications:
  - Balance A degrades with increasing batch sizes
  - Soil quality deviation σ varies from 30% to 80%, since balance is not satisfied
  - Running time inversely proportional to batch size
- Fixed time (10 min.):
  - Number of replications proportional to batch size
  - Optimal batch size: 256, with best balance $A = 23.3$ and $\sigma = 31.8$

Batch	Fixed time				50 replications			
size	A (m)	λ	σ	Repl.	A (m)	λ	σ	t (s)
32	40.3	4.3	44.6	2,494	28.7	4.0	35.7	668
64	28.4	3.8	27.6	6,118	35.5	3.8	29.3	301
128	28.4	3.7	32.0	13,760	34.4	4.3	36.7	120
256	23.3	4.1	31.8	27,102	44.2	5.5	42.0	52
512	28.5	4.5	44.5	45,790	55.8	4.0	54.7	26
1,024	42.2	4.2	41.5	67,831	75.6	4.8	56.8	15
2,048	51.1	4.5	65.4	88,424	94.3	6.8	58.8	10
4,096	67.9	6.4	82.4	111,203	118.7	6.3	81.6	8

EXPERIMENT 2: SCALABILITY

Size	Lots	Evals.	A (μ)	λ	σ (K)	n
200x200	20	138,732	0.00	1.70	5.66	5
	40	79,999	0.00	1.75	1.83	5
	60	56,350	0.00	1.81	1.78	5
	80	43,927	0.00	1.84	1.38	5
	100	37,721	436.24	5.39	5.10	0
300x300	20	60,013	0.00	3.40	81.19	1
	40	31,882	0.00	1.90	10.40	5
	60	22,982	0.00	1.88	6.93	5
	80	17,937	0.00	1.91	5.33	5
	100	15,214	0.00	2.11	4.70	5
400x400	20	22,761	0.00	1.29	26.83	5
	40	13,254	0.00	1.43	12.93	5
	60	8,923	0.00	1.46	7.83	5
	80	7,135	0.00	1.51	5.75	5
	100	6,237	0.00	1.57	7.07	5
500x500	20	9,970	0.00	2.39	66.52	5
	40	5,704	0.00	2.62	32.37	5
	60	4,058	4.12	6.17	28.19	4
	80	3,143	0.00	2.73	18.57	5
	100	2,714	0.00	2.81	14.54	5
600x600	20	7,502	0.00	1.89	297.75	5
	40	5,384	0.00	2.20	128.50	5
	60	4,125	11.10	2.45	82.63	4
	80	3,396	35.63	3.45	58.65	3
	100	2,233	0.00	2.54	47.35	5

EXPERIMENT 2: SCALABILITY

Size	Lots	Evals.	A (μ)	λ	σ (K)	n
	20	138,732	0.00	1.70	5.66	5
	40	79,999	0.00	1.75	1.83	5
200×200	60	56,350	0.00	1.81	1.78	5
	80	43,927	0.00	1.84	1.38	5
	100	37,721	[illegible]	[illegible]	[illegible]	[illegible]
	20	60,012	0.00	[illegible]	81.19	1
	40	31,882	0.00	1.90	10.40	5
300×300	60	22,982	0.00	1.88	6.93	5
	80	17,937	0.00	1.91	5.33	5
	100	15,214	0.00	2.11	4.70	5
	20	22,761	0.00	1.29	26.83	5
	40	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
400×400	60	8,923	0.00	1.46	7.83	5
	80	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
	100	6,237	0.00	1.57	7.07	5
	20	9,970	0.00	2.39	66.52	5
	40	5,704	0.00	2.62	32.37	5
500×500	60	4,058	4.12	6.17	28.19	4
	80	3,143	0.00	2.73	18.57	5
	100	2,714	0.00	2.81	14.54	5
	20	7,502	0.00	1.89	297.75	5
	40	5,384	0.00	2.20	128.50	5
600×600	60	4,125	11.10	2.45	82.63	4
	80	3,396	35.63	3.45	58.65	3
	100	2,233	0.00	2.54	47.35	5

- Feasible solutions in 112 of the 125 tests
- No feasible solution for 200×200 with 100 lots: hard to satisfy balance when there are too many lots in a small area
- As expected, number of fitness evaluations decreases with instance size and number of lots: about $2.1 \times 10^{12}\, n^{-2.7}\, k^{-0.8}$ (log-log regression, $R^2 = 0.9812$)

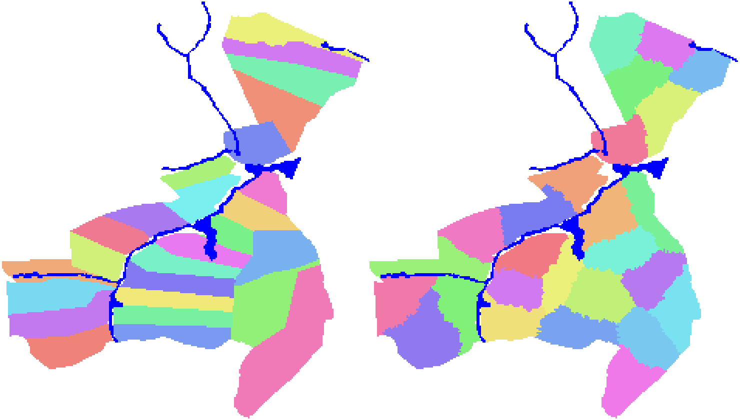
EXPERIMENT 3: EFFECTIVENESS OF THE GA

Instance	Fitness evals. ($\times 10^4$)			A (m)			λ			σ (% r.d.)		
	BFS	Cons.	GA	BFS	Cons.	GA	BFS	Cons.	GA	BFS	Cons.	GA
Belo Vale	65.9	2.0	14.4	0.1	0.0	0.0	9.99	2.99	1.83	657.4	196.6	5.9
Fortaleza	83.2	2.4	16.2	38.7	25.1	0.0	59.79	4.86	1.19	1,025.7	479.4	4.7
Iucatã	62.9	3.6	34.5	0.0	64.8	0.0	10.26	18.59	2.83	582.9	463.0	63.0
Olhos D'Água	75.5	5.2	59.8	0.0	82.7	0.0	5.67	8.97	1.93	165.2	54.8	9.7
Veredas	146.1	2.6	23.4	1.9	49.0	0.0	22.62	3.87	1.70	424.8	95.7	12.1

- Repeated BFS found solutions with lower balance violations than the greedy constructive approach in 4 of the 5 instances, but with higher violations of equality and much worse quality deviations
- GA found feasible solutions for all 5 instances w.r.t. balance and equality, and considerably lower quality deviations
- Indicates that recombination and mutation are effective: GA is much better than just repeated greedy construction

Instance	A (m)		λ		σ	
	Manual	GA	Manual	GA	Manual	GA
Belo Vale	19.7	0.0	2.77	1.83	5,378.5	1,888.6
Iucatã	735.9	0.0	22.38	2.83	18,444.9	2,806.5
Olhos D'Água	830.5	0.0	28.81	1.93	19,972.8	11,716.1
Veredas	0.0	0.0	5.78	1.70	3,964.4	1,161.5

- GA is better w.r.t. all three components: equality, balance and soil quality deviation
- Manual solutions violate either balance or equality in all instances, while GA finds feasible solutions
- Soil quality deviation found by GA is a factor of 1.7 to 7 lower



Solutions of real-world instance "Veredas".
Left: manually produced by INCRA. Right: produced by the GA.

Solutions to real-world instance “Iucatã” (Acre).
Left: manually produced by INCRA. Right: generated by the GA.

Conclusions

CONCLUSIONS

- GA scales reasonably well with instance size and number of lots
- GA is effective compared to simpler methods and compared to manual allocation
- Potential for better solutions than those currently applied in practice

THANKS FOR YOUR ATTENTION!

Alex Gliesch, Marcus Ritt, Mayron Moreira
Instituto de Informática — UFRGS
inf.ufrgs.br