

META-HEURISTIC CLONAL SELECTION ALGORITHM FOR OPTIMIZATION OF FOREST PLANNING¹

Carlos Alberto Araújo Júnior^{2*}, João Batista Mendes³, Christian Dias Cabacinha², Adriana Leandra de Assis², Lisandra Maria Alves Matos⁴ and Helio Garcia Leite⁵

¹ Received on 26.06.2017 accepted for publication on 22.11.2017.

² Universidade Federal de Minas Gerais, Professor do Curso de Engenharia Florestal, Montes Claros, Minas Gerais – Brasil. E-mail: <carlosaraujo@ufmg.br>, <christian.cabacinha@gmail.com> and <adrianaleandra@gmail.com>.

³ Universidade Estadual de Montes Claros, Departamento de Computação, Minas Gerais – Brasil. E-mail: <joao.mendes@unimontes.br>.

⁴ Universidade Federal de Minas Gerais, Graduada de Engenharia Florestal, Montes Claros, Minas Gerais – Brasil. E-mail: <lisandraamatos@gmail.com>.

⁵ Universidade Federal de Viçosa, Departamento de Engenharia Florestal, Viçosa, Minas Gerais – Brasil. E-mail: <hglete@gmail.com>.

*Corresponding author.

ABSTRACT – It is important to evaluate the application of new technologies in the field of computational science to forest science. The goal of this study was to test a different kind of metaheuristic, namely Clonal Selection Algorithm, in a forest planning problem. In this problem, the total management area is 4.210 ha that is distributed in 120 stands in ages between 1 and 6 years and site indexes of 22 m to 31 m. The problem was modeled considering the maximization of the net present value subject to the constraints: annual harvested volume between 140,000 m³ and 160,000 m³, harvest ages equal to 5, 6 or 7 years, and the impossibility of division of the management unity at harvest time. Different settings for Clonal Selection Algorithm were evaluated to include: varying selection, cloning, hypermutation, and replacement rates beyond the size of the initial population. A generation value equal to 100 was considered as a stopping criteria and 30 repetitions were performed for each setting. The results were compared to those obtained from integer linear programming and linear programming. The integer linear programming, considered to be the best solution, was obtained after 1 hour of processing. The best setting for Clonal Selection Algorithm was 80 individuals in the initial population and selection. Cloning, hypermutation, and replacement rates equal to 0.20, 0.80, 0.20 and 0.50, respectively, were found. The results obtained by Clonal Selection Algorithm were 1.69% better than the integer linear programming and 4.35% worse than the linear programming. It is possible to conclude that the presented metaheuristic can be used in the resolution of forest scheduling problems.

Keywords: Operational research; Artificial intelligence; Artificial immunological system.

METAHEURÍSTICA CLONAL SELECTION ALGORITHM PARA OTIMIZAÇÃO DO PLANEJAMENTO FLORESTAL

RESUMO – Dada a importância de se avaliar novas tecnologias para otimização do planejamento florestal, este trabalho objetivou introduzir a metaheurística *Clonal Selection Algorithm* na resolução de um problema de ordenamento da produção florestal. Considerou-se uma área manejada de tamanho igual a 4.210 ha, contendo 120 talhões com idades entre 1 e 6 anos e índice de sítio variando entre 22 m e 31 m. O problema foi modelado com o objetivo de se maximizar o valor presente líquido global do empreendimento e considerou como restrições uma demanda anual entre 140.000m³ e 160.000 m³, colheita apenas nas idades de 5, 6 e 7 anos e a imposição de não fracionamento dos talhões no momento do corte. Foram avaliados diferentes configurações da metaheurística *Clonal Selection Algorithm*, variando-se as taxas de seleção, clonagem, hipermutação e substituição, além do tamanho da população inicial. Considerou-se como critério de parada uma quantidade de gerações igual a 100 e para cada parametrização avaliou-se 30 repetições. Os resultados foram comparados com aqueles obtidos utilizando-se programação linear e programação linear inteira. Para a programação inteira considerou-se a melhor solução após 1 hora de processamento. A melhor configuração



foi a que considerou 80 células na população inicial, taxas de seleção, clonagem, hipermutação e substituição iguais a 0,20, 0,80, 0,20 e 0,50, respectivamente. Os resultados apresentados pela metaheurística Clonal Selection Algorithm foram 1,69% superiores à programação linear inteira e 4,35% inferior a programação linear. Conclui-se que a metaheurística apresentada pode ser utilizada para resolução de problemas de ordenamento florestal.

Palavras-Chave: Pesquisa operacional; Inteligência artificial; Sistemas imunológicos artificiais.

1. INTRODUCTION

Hierarchical forest planning problems must consider decisions about activities that will be done in time and space. These decisions are necessary to maintain a constant flow of wood, obtain the maximum economic return, and improve ecological and social aspects, among other objectives (Kaya et al., 2016; Ezquerro et al., 2016; Dong et al., 2016). These choices must be made considering an optimization process at all hierarchical levels in the decision making processes (strategic, tactical, and operational). To do so, it is important to select the best set of actions, or make the best decisions, for a given problem that considers limited resources (Kaya et al., 2016). However, the number of decision variables and constraints considered in the modeling is too wide in most situations. Some of these constraints can be singular and it is difficult to obtain an optimal solution using exact algorithms. This prevents variable values from being obtained in an intuitive way and considers only the planner's expertise or is performed by a trial and error method.

Mathematical models for forest planning can be formulated as a Linear Programming problem (LP), a Linear Integer Programming problem (IP) or a Mixed Integer Linear Programming problem (MIP) when the problems are relatively simple (Troncoso et al., 2016). Additionally, they can be solved using deterministic algorithms such as Simplex, and Branch and Bound (Hillier and Lieberman, 2013).

These algorithms are recommended when the problems are considered to belong to the P class (*polynomial time*). Problems in this class can be solved with deterministic methods in a brief time, or in a polynomial time. Nondeterministic algorithms are used when the problems are considered as NP (nondeterministic polynomial time). This suggests the need for further development and application of heuristics methods (Yoshimoto et al., 2016; Jin et al., 2016).

A heuristic is an iterative method that searches good feasible solutions for a problem but does not ensure the attainment of a global solution (Hillier and

Lieberman, 2013; Jin et al., 2016; Ezquerro et al., 2016). Heuristics algorithms are formulated to explore a wide area in the solutions space. They do it in a random or semi-random way (Yoshimoto et al., 2016). Their primary advantage is their flexibility, because they can be used to solve problems in which additivity and proportionality requirements are not feasible (Jin et al., 2016). However, they can be susceptible to local optimum problems because they do not have the ability to escape from these regions changing drastically the local search. They also are specific for a problem and require a distinct model for each situation.

Hence, heuristic algorithms that have an ability to explore different regions of the solution space were developed and are called metaheuristics. They are techniques based on general operations applicable to most parts of combinatorial problems (Gaspar-Cunha et al., 2013). They are found in a wide area of metaheuristics (Boussaid et al., 2013) to include the following: Tabu Search (TS), Simulated Annealing (SA), Genetic Algorithm (GA), Ant Colony (AC), GRASP and HERO, Particle Swarm Optimization (PSO), Bee Colony (BC), Multiagent Systems (MAS) and Artificial Neural Networks (ANN). Also, there have been many applications in planning and forest science developed since 1980 (Jin et al., 2016; Ezquerro et al., 2016). The number of papers regarding forest planning problems solved by metaheuristics have already surpassed the number of scientific works that use classical methods in operation research (Ezquerro et al., 2016). This has occurred due to the fact that metaheuristics have potential use in complex forest management plans (Dong et al., 2016). Simulated annealing is the most cited method in the literature (Ezquerro et al., 2016).

Other metaheuristic methods have been developed to support recent studies in artificial intelligence. It allows the application of these methods to be applied throughout the field of operation research. A set of algorithms that are emerging in combinatorial problems is the Artificial Immune Systems (AIS). They are bio-inspired algorithms based on natural immune system behavior whose main function is to detect and protect the organism against foreign organisms that can be

a cause of diseases (Sreeram and Panicker, 2015; Bernardino et al., 2011). Among them, Clonal Selection Algorithm (CSA), or Clonalg, is the most used algorithm in optimization problems (Wang et al., 2016; Riff et al., 2013). This algorithm is based on clonal selection and affinity maturation principles (Liu et al., 2010; Boussaid et al., 2013).

In this study we evaluated the metaheuristic CSA behavior considering different sets of parameters values and their relative efficiency compared to Linear Programming and Linear Integer Programming models in solving a forest planning problem.

2. MATERIAL AND METHODS

Case study

A eucalyptus forest with 4.210 ha divided into 120 stands (management units) was considered in this study. The distribution of area per forest age is irregular: 339 ha with 1-year-old, 768 ha with 2-year-old, 1031 ha with 3-year-old, 601 ha with 4-year-old, 958 ha with 5-year-old, and 513 ha with 6-year-old forestry. Production estimates of wood for each stand and each age were obtained from the equation $V_i = 6,09 - 117,55 I_i^{-1} S_i^{-1}$, where V_i is the volume for the stand i , I_i is the age of stand i , and S_i is the site index for the stand i . These estimates were developed by the authors considering data from a continuous forest inventory. Values of the site index ranged from 22 m to 31 m, and had an index age equivalent to six years.

Management prescriptions for each stand considered the possibility of harvest in one of three ages (five, six, and seven years) with immediate planting activity and a time horizon of the forest plan equal to sixteen years. Thus, 81 different prescriptions for each of the 120 stands were evaluated. These prescriptions demanded 9720 decision variables. The Model I structure was considered to maximize the net present value (NPV) with constraints that assured an annual production of wood between 140,000 m³ and 160,000 m³.

The annual discount rate used was equal to 8 percent, the price of wood sales equal to R\$ 80.00 per cubic meter, and harvester cost equal to R\$ 30.00 per cubic meter. Silvicultural costs ranged according to the age of each stand and were obtained from Binoti (2010): R\$ 4.059,05 ha⁻¹ on first-year; R\$ 1.627,81 ha⁻¹ on

second-year; R\$ 757,95 ha⁻¹ on third-year; and R\$ 88,12 ha⁻¹ on the fourth-year of forest growth until harvest age.

Linear Programming and Linear Integer Programming

LP and IP models were formulated as suggested by Rodrigues et al. (2004). A unique difference between these models is the absence of constraints 2 and 5 in the LP model:

$$\text{Max GNPV} = \sum_{i=1}^M \sum_{j=1}^N C_{ij} X_{ij} \quad (\text{Eq- 1})$$

Subject to

$$\sum_{i=1}^M \sum_{j=1}^N X_{ij} = 1 \quad (\text{Eq- 2})$$

$$\sum_{i=1}^M \sum_{j=1}^N V_{ij(k)} X_{ij} \geq D_{\text{min}_k} \quad (\text{Eq- 3})$$

$$\sum_{i=1}^M \sum_{j=1}^N V_{ij(k)} X_{ij} \leq D_{\text{max}_k} \quad (\text{Eq- 4})$$

$$X_{ij} \in \{0,1\} \quad (\text{Eq- 5})$$

where $GNPV$ is the global net present value for all the forest in reais; C_{ij} is NPV for stand i when assigned the prescription j in reais; X_{ij} is the decision variable and represents the proportion area of stand i that will be managed with prescription j ; M is the total number of stands; N is the total number of different prescriptions for each stand; $V_{ij(k)}$ is the total volume of wood for the stand i , when assigned the prescription j , in the period k of planning horizon; D_{min_k} and D_{max_k} are minimum and maximum wood demand for the period k of the planning horizon.

Constraint (2) guarantees that all stand areas receive one prescription. Constraints (3) and (4) limit the annual harvested volume between minimum and maximum values of demand, and constraint (5) imposes that there is only one management prescription for each stand. This last constraint was applicable to the IP model.

LP and IP models were solved using the Lingo software and the Simplex and Branch and Bound algorithms, respectively. A computer with a Windows 10 operating system, 64-bit, processor Intel Core i7 with 2.0 GHz and 8 GB of ram memory was used.

Clonal Selection Algorithm

The CSA algorithm is inspired by the natural process of antibodies clonal selection. This process considers that only cells that recognize a specific antigen must be proliferated (Wang et al., 2016). During proliferation, these cells suffer a somatic hypermutation, that can be gradually improved in their ability to recognize the antigen and make the immune response more effective (Qiu and Lau, 2014). One generation of cells in this algorithm, or one iteration in the search process, includes the candidate solutions formulation, selection, cloning, mutation, and a new selection and a change of the initial population. All these aspects are related to the evolutionary algorithms (Boussaid et al., 2013).

The CSA starts its search process by creating a pre-determined number of antibodies which represent possible solutions to the problem when applied to combinatorial problems (Figure 1). The antibodies of initial population are evaluated in terms of a problem objective function value (as represented by Equation 1 presented for the linear programming problem) and a set of best-adapted antibodies which is selected according to a selection rate. Next, these antibodies are cloned and suffer a hypermutation process to generate cells with a better affinity to the antigen, in other words, to generate better solutions to the problem.

The best clones of the new population are selected to compose the next population which will substitute the previous population by exclusion of less adapted individuals. According to Castro and Von Zuben (2002), the number of generated clones for each selected antibody is calculated by:

$$n_c(ac_i) = (\beta ac_i) \quad (\text{Eq-6})$$

where n_c is the number of clones for the selected antibody, β is the cloning rate, and ac_i is the total number of antibodies in the initial population.

Considering a hypermutation rate (ρ), a Castro and Von Zuben (2002) and Wang et al. (2016) equation was used to calculate the number of mutations for each clone:

$$a(c_i) = \exp(-\rho^* f_i) \quad (\text{Eq-7})$$

where α is the number of mutations for clone i , f_i is the normalized fitness function value and ρ^* is hypermutation parameter calculated considering the hypermutation rate of the system:

$$\rho^* = 5(1 - \rho) \quad (\text{Eq-8})$$

Equation 8 was used to normalize the input value of the hypermutation rate between 0 and 1. This becomes easier to understand by instead considering an interval between 5 (absence of hypermutation) and 0 (total hypermutation).

No literature exists that evaluates the CSA in forest problems, thus it was necessary to test different arrangements of its parameters in order to obtain a setup that generates satisfactory results. In this way, different values for the hypermutation rate (test 1), cloning rate (test 2), selection rate (test 3), replacement rate (test 4) and initial population size (test 5) were considered.

Three values for hypermutation, cloning, selection and replacement rates were evaluated: 0.20, 0.50 and 0.80, respectively. Three values for initial population size were considered: 20, 50 and 80 individuals, respectively. The identification of the optimal setup was done in steps with each one representing one parameter selection. Firstly, different values for hypermutation rates were considered, maintaining other parameters fixed. The cloning rate was evaluated with the best hypermutation rate with all other parameters fixed. Next, the best selection rate, replacement rate and initial population size were defined.

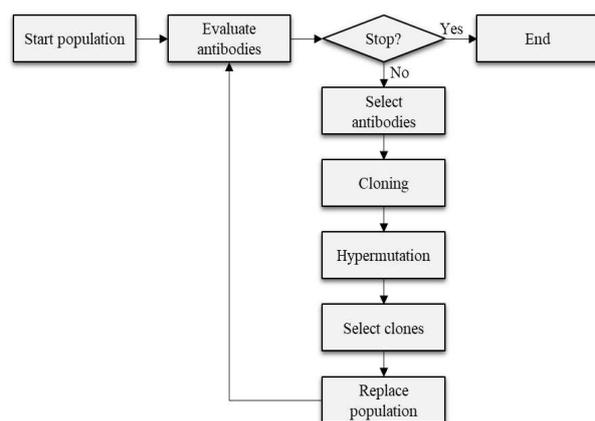


Figure 1 – Flowchart of processing routine of Clonal Selection Algorithm. Adapted from Castro and Von Zuben (2002).

Figura 1 – Fluxograma da rotina de processamento do Clonal Selection Algorithm. Adaptado de Castro and Von Zuben (2002).

Each evaluation considered 30 repetitions and the stopping criteria was defined as a number of iterations equal to 100. The configuration that obtained the optimal values for the average of fitness, maximum fitness, and a small value of standard deviation at end of 100 generations was chosen.

Identical constraints imposed on the integer linear programming model were also considered (minimum and maximum annual harvested volume and only one prescription for each management unit). The objective function was penalized when a constraint was broken. Thus, the fitness function was decreased by R\$ 100.00 for each cubic meter of wood above or over the demand minimum and maximum in each year of the planning horizon. This method is the same adopted by Rodrigues et al. (2004).

Metaheuristic processing was performed using MeP (*Metaheuristics for Forest Planning*) software developed in the Java language at Operations Research and Forest Modeling Laboratory of the Federal University of Minas Gerais.

3. RESULTS

The CSA processing finished when the stopping criteria was reached and the processing time varied according to the parameters values used. For an initial population size of 20 cells, the average processing time was equal to 33 seconds for each repetition (with a minimum

equal to 32 s and a maximum equal to 36 s), for 50 cells the value was 81 seconds (with a minimum equal to 80 s and a maximum equal to 84 s) and for 80 cells the value was 139 seconds (with a minimum equal to 134 s and maximum equal to 151 s).

The best CSA parametrization considered an initial population size equal to 80 cells, a replacement rate equal to 0.50, a selection rate equal 0.2, a cloning rate equal to 0.80, and a hypermutation rate equal to 0.2 (Table 1).

Considering the results of CSA, there was a quick convergence of solutions to values near the final value found, except for cases in which greater hypermutation rates were used (Figure 2). When the algorithm exhibits an increase in the cloning rate, it converges faster to the best solution with a less number of generations. There were no observed drastic behavior differences when the values of replacement and selection rates were changed. The increase in the initial population size allowed the algorithm to reach optimal values with few iterations and a minor value of standard deviation.

The Linear Integer Programming solution was obtained after 1 hour of processing, but the algorithm did not converge to the global optimal solution; on the other hand, Linear Programming identified the global optimal solution. The CSA solution was a better solution than the one identified by IP, 3.54% greater. The CSA solution was identified to be 4.35% inferior to the LP solution.

Table 1 – Results from evaluations considering different parameters of Clonal Selection Algorithm.

Tabela 1 – Resultados das avaliações realizadas com diferentes parâmetros para o Clonal Selection Algorithm.

Test	IPS	RER	SER	CLR	HYR	NGE	Maximum of fitness (R\$)	Minimum of fitness (R\$)	Average of fitness (R\$)	Standard deviation of fitness (R\$)
1	20	0.2	0.2	0.2	0.2*	100	29,679,553	30,441,581	30,125,437	193,602
	20	0.2	0.2	0.2	0.5	100	25,024,349	29,762,745	27,788,508	1,442,727
	20	0.2	0.2	0.2	0.8	100	4,482,727	20,582,717	12,403,374	4,287,838
2	20	0.2	0.2	0.2	0.2	100	29,679,553	30,441,581	30,125,437	193,602
	20	0.2	0.2	0.5*	0.2	100	30,065,528	30,627,268	30,340,494	144,735
	20	0.2	0.2	0.8	0.2	100	29,986,526	30,579,158	30,337,133	148,248
3	20	0.2	0.2	0.5	0.2	100	30,065,528	30,627,268	30,340,494	144,735
	20	0.2	0.5	0.5	0.2	100	30,065,499	30,603,016	30,343,623	159,285
	20	0.2	0.8*	0.5	0.2	100	29,879,122	30,727,585	30,363,336	181,738
4	20	0.2*	0.8	0.5	0.2	100	29,879,122	30,727,585	30,363,336	181,738
	20	0.5	0.8	0.5	0.2	100	30,010,931	30,665,962	30,302,277	142,759
	20	0.8	0.8	0.5	0.2	100	30,021,089	30,498,711	30,293,191	122,057
5	20	0.2	0.8	0.5	0.2	100	29,879,122	30,727,585	30,363,337	181,738
	50	0.2	0.8	0.5	0.2	100	30,197,987	30,660,003	30,467,552	127,292
	80*	0.2	0.8	0.5	0.2	100	30,403,942	30,791,495	30,565,538	106,559

Where: IPS is the initial population size; RER is the replacement rate; SER is the selection rate; CLR is the cloning rate; HYR is the hypermutation rate; e NGE is the number of generations.

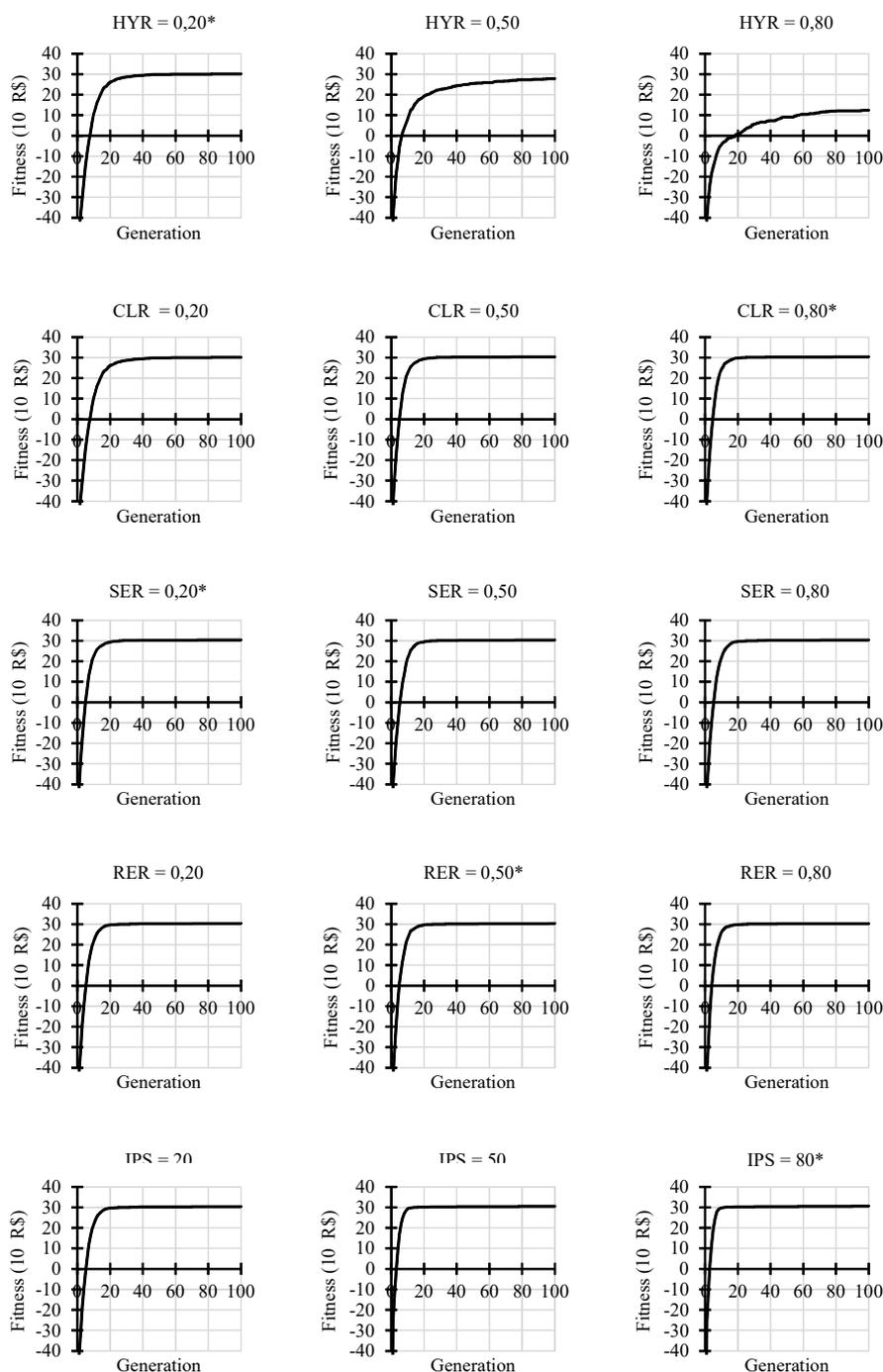


Figure 2 – Evolution of solutions average for each iteration of Clonal Selection Algorithm considering different settings. HYR, CLR, SER and RER are hypermutation, cloning, selection and replacement rates. IPS is the initial population size.

Figura 2 – Evolução das médias das soluções obtidas a cada iteração do Clonal Selection Algorithm considerando diferentes configurações. Onde: HYR, CLR, SER e RER representam as taxas de hipermutação, clonagem, seleção e substituição; IPS é o tamanho da população inicial.

Table 2 – Best solutions results found by each method and its proportion in relation to linear programming (LP) and integer linear programming (IP).**Tabela 2** – Resultados das melhores soluções encontradas por cada método e sua proporção em relação ao resultado da Programação Linear (LP) e Programação Linear Inteira (IP).

Method	Best solution value	% in relation to LP	% in relation to IP
Linear Programming	R\$ 32,191,790	100.00%	106.31%
Integer Linear Programming	R\$ 30,279,900	94.06%	100.00%
Clonal Selection Algorithm	R\$ 30,791,495	95.65%	101.69%

The LP solution did not obtain binary values for all decision variables. This fact occurred in 25 of them (0.26%). These values were distributed in 12 of 120 stands. One stand received three different prescriptions and the other eleven received two prescriptions.

The annually harvested volume constraint was attempted in the best identified solution for all methods. Differences were observed in relation to the period when more or less wood volume is harvested (Figure 3). Also, there were differences in the total volume produced and in the variation of harvester along two consecutive years. Total volume obtained with the best solutions were: 2.265.710 m³ for LP; 2.387.331 m³ for IP; and 2.339.034 m³ for CSA. Minimum and maximum annual harvested volume were: 140.000 m³ and 158.058 m³ for LP; 140.598 m³ and 159.209 m³ for IP; and 140.435 m³ and 157.951 m³ for CSA. Maximum variations in the harvested volume from one year to another were: 12,90% for LP; 13,24% for IP; e 6,79% for CSA.

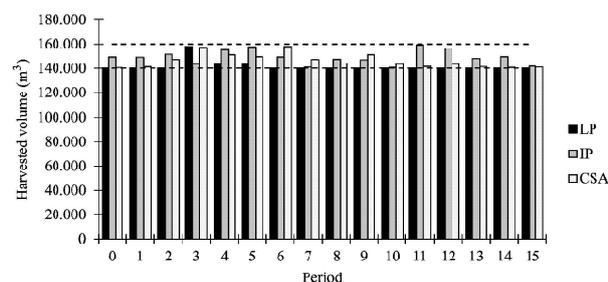
**Figure 3** – Evolution of annual cut volume considering the best solutions found by linear programming (LP), integer linear programming (IP) and Clonal Selection Algorithm (CSA). The dashed lines indicate the values of maximum and minimum annual demand.

Figura 3 – Evolução do volume de colheita anual considerando as melhores soluções encontradas para a Programação Linear (LP), Programação Linear Inteira (IP), Clonal Selection Algorithm (CSA). As linhas tracejadas indicam os valores de demanda máxima e mínima anual.

4. DISCUSSION

Strategic forest planning problems that aim to create a harvest prescription at a stand level are recognized by their high complexity. This suggests a need for the usage of loss impeditive mathematical programming models, such as linear programming. The results show that it is possible to realize LP generating an optimal solution with an objective function value greater than other methods (Silva et al., 2003).

However, this solution assumes that it will be necessary to adopt different prescriptions for a single stand. This causes a subdivision of the management unit because the harvest operation can be applied in different periods. Considering that is impossible to realize a harvest operation in only a stand fragment, in most cases, it is necessary to impose an integrity constraint in the mathematical model. This is necessary because rounding the solutions from linear programming is not recommended in these cases (Silva et al., 2003). Thus, the LP model results presented must be considered as an informational value to compare with the other methods that consider the adoption of only one prescription for each management unit. According to Dong et al. (2016), LP results can be considered as performance evaluation criteria for metaheuristics.

It is more difficult to obtain a solution that attempts other constraints, such as minimum and maximum annual harvested volume, when stand integrity constraint is adopted. Therefore, for the three methods evaluated, the forest production flow along time maintained between 140,000 m³ and 160,000 m³. However, there was a difference in the total harvested volume obtained in the horizon planning and a variability in the amount of wood harvested along two consecutive years.

Linear programming presented the greatest value for global net present value, although it presented a minor total harvested volume. This can be explained by an optimal definition of harvest areas along years

in a horizon plan, once this model is more flexible. The CSA solution generated a minor variation in the harvested volume in two succeeding periods. This is desirable in practical cases where the structure (workers and machines, for instance) necessary in harvest operations cannot be drastically changed from one year to another. It is important to emphasize that a constraint considering the variation of harvested volume in two succeeding years was not modeled.

CSA results were superior to the IP results. This highlights the importance of studies utilizing new algorithms to solve the same problem. Indeed, CSA has been presented satisfactory results in different optimization problems, as mentioned by Sreeram and Panicker (2015) and Bernardino et al. (2011). In this way, according to Yoshimoto et al. (2016), if the quality of a solution is considered high when compared to a determinist technique, then the heuristic method is preferable. Yet, when well structured, heuristics can provide near optimal solutions to complex problems (Kaya et al., 2016).

Usually, metaheuristic methods do not find a global optimum solution. However, this is not a problem since there is a high level of uncertainty in volume estimates, prices forecasts, pest occurrence, diseases and others abiotic factors that affect the forest planning (Jin et al., 2016).

Another highlighted result is related to the time spent for CSA to gain good solutions. The best one was found in only 3 minutes while IP spent 60 minutes to find an inferior solution to the CSA method. Indeed, in combinatorial problems, time is an important factor and is sometimes used as an algorithm quality indicator. The time necessary to obtain a good and feasible solution in forest planning is additionally very important because analysts must generate different planning scenarios in a short amount of time. This is done to allow the manager to elect one of these scenarios and directs the forest investments along a strategical plan. Thus, quick algorithms that generate good solutions that attempt all constraints in a short amount of time must be considered and prioritized in practical applications.

Parameter values affect directly the metaheuristic performance, as observed by Rodrigues et al. (2004) and Jin et al. (2016). In CSA, the hypermutation rate parameter presented results with the greatest variability in our evaluation. As high as this parameter was, the

worse was the algorithm behavior. This is explained by the fact that too many high hypermutation rates can create randomly new solutions in practice and eliminate solutions that had presented good values of fitness. High rates can also cause drastic mutations in the best solutions in each generation, mainly because the mutations suffered by the antibodies are inversely proportional to their fitness value (Qiu and Lau, 2014).

According to Boussaid et al. (2013), mutation operators work as a local optimum escaped operator in evolutionary algorithms. Thus, it is necessary to calibrate mutation rates to guarantee that only the interested solution neighborhood be explored. Aggressive mutations, which change too many gene values, can make the search randomized, and this is not desirable.

The final solution of CSA did not present wide variations neither in relation to cloning, selection, and replacement rates nor in relation to the size of the initial population. The increase in the cloning rate reached good solutions with very few generations; that is desirable. The proliferation of antibodies is directly proportional to the affinity of the antigen (Qiu and Lau, 2014) and this phenomenon is called affinity maturation. In other words, the better a solution is, the more time the antibody is cloned, which improves the search by solutions each time. This also occurred when the initial population size was increased.

Cloning rate and the number of antibodies are associated with the number of evaluated solutions by generation. As high it is, even higher so will be the search in the solutions space. This increases the possibility of finding good solutions with few iterations. However, if the stopping criteria is the number of generations, a high cloning rate causes loss of performance in terms of processing time, mainly, when the initial population size is very large. Thus, it is always necessary to evaluate these parameters values to choose ones that can obtain good results conciliating generation number and processing time.

5. CONCLUSION

According to the results obtained in this study, it is possible to conclude that CSA is efficient when used to solve combinatorial optimization problems in forest planning. Moreover, to choose the best set of parameters it is crucial to get good solutions with less computational efforts.

6. ACKNOWLEDGMENT

The authors are grateful to Fundação de Amparo à Pesquisa do Estado de Minas Gerais - FAPEMIG and to Universidade Federal de Minas Gerais - UFMG by the financial and technical support.

7. REFERENCES

- Bernardino HS, Barbosa HJC, Fonseca LG. Surrogate-assisted clonal selection algorithms for expensive optimization problems. *Evol Intell.* 2011;4(1):81-97. <http://dx.doi.org/10.1007/s12065-011-0056-1>.
- Binoti DB. Estratégias de regulação de florestas equiâneas com vistas ao manejo da paisagem [dissertação]. Viçosa: Universidade Federal de Viçosa; 2010.
- Boussaid I, Lepagnot J, Siarry P. A survey on optimization metaheuristics. *Inf Sci.* 2013;237(1):82-117. <http://dx.doi.org/10.1016/j.ins.2013.02.041>.
- Castro LN, Von Zuben FJ. Learning and optimization using the clonal selection principle. *IEEE Trans Evol Comput.* 2002;6(3):239-51. <http://dx.doi.org/10.1109/TEVC.2002.1011539>.
- Dong L, Bettinger P, Liu Z, Qin H, Zhao Y. Evaluating the neighborhood, hybrid and reversion search techniques of a simulated annealing algorithm in solving forest spatial harvest scheduling problems. *Silva Fenn.* 2016;50(4):1-20. <http://dx.doi.org/10.14214/sf.1622>.
- Ezquerro M, Pardos M, Diaz-Balteiro L. Operational research techniques used for addressing biodiversity objectives into forest management: an overview. *Forests.* 2016;7(10):229-47. <http://dx.doi.org/10.3390/f7100229>.
- Gaspar-Cunha A, Takahashi R, Antunes CH. Manual de computação evolutiva e metaheurísticas. 1. ed. Belo Horizonte: Editora UFMG; 2013.
- Hillier FS, Lieberman GJ. Introdução à pesquisa operacional. 9. ed. Porto Alegre: AMGH; 2013.
- Jin X, Pukkala T, Li F. Fine-tuning heuristic methods for combinatorial optimization in forest planning. *Eur J For Res.* 2016;135(1):765-79. <http://dx.doi.org/10.1007/s10342-016-0971-x>.
- Kaya A, Bettinger P, Boston K, Akbulut R, Ucar Z, Siry J, et al. Optimisation in Forest Management. *Curr For Rep.* 2016;2(1):1-17. <http://dx.doi.org/10.1007/s40725-016-0027-y>.
- Liu R, Jiao L, Li Y, Liu J. An immune memory clonal algorithm for numerical and combinatorial optimization. *Front Comput Sci China.* 2010;4(4):536-59. <http://dx.doi.org/10.1007/s11704-010-0573-6>.
- Qiu X, Lau HYK. An AIS-based hybrid algorithm for static job shop scheduling problem. *J Intell Manuf.* 2014;25(1):489-503. <http://dx.doi.org/10.1007/s10845-012-0701-2>.
- Riff MC, Montero E, Neveu B. Reducing calibration effort for clonal selection based algorithms: a reinforcement learning approach. *Knowl Base Syst.* 2013;41(1):54-67. <http://dx.doi.org/10.1016/j.knosys.2012.12.009>.
- Rodrigues FL, Leite HG, Santos HN, Souza AL, Silva GF. Metaheurística algoritmo genético para solução de problemas de planejamento florestal com restrições de integridade. *Rev Arvore.* 2004;28(2):233-45. <http://dx.doi.org/10.1590/S0100-67622004000200010>.
- Silva GF, Leite HG, Silva ML, Rodrigues FL, Santos HN. Problemas com o uso de programação linear com posterior arredondamento da solução ótima, em regulação florestal. *Rev Arvore.* 2003;27(5):677-88. <http://dx.doi.org/10.1590/S0100-67622003000500010>.
- Sreeram KY, Panicker VV. Clonal selection algorithm approach for multi-objective optimization of production-distribution system. *Procedia Soc Behav Sci.* 2015;189(1):48-56. <http://dx.doi.org/10.1016/j.sbspro.2015.03.191>.
- Troncoso JJ, Weintraub A, Martell DL. Development of a threat index to manage timber production on flammable forest landscapes subject to spatial harvest constraints. *Inf Syst Oper Res.* 2016;54(3):262-81. <http://dx.doi.org/10.1080/03155986.2016.1197543>.
- Wang X, Deshpande AS, Dadi GB, Salman B.

Application of Clonal Selection Algorithm in construction site utilization planning optimization. *Procedia Eng.* 2016;145(1):267-73. <http://dx.doi.org/10.1016/j.proeng.2016.04.073>.

Yoshimoto A, Asante P, Konoshima M. Stand-level forest management planning approaches. *Curr For Rep.* 2016;2(3):163-76. <http://dx.doi.org/10.1007/s40725-016-0041-0>.