In **Farming Systems Design 2007: Methodologies for Integrated Analysis of Farm Production Systems**, (Donatelli, M., Hatfield, J., and Rizzoli, A., Eds.), La Golliardica Pavese, September 2007, pp.225-226.

GENERATION OF ARTIFICIAL CROP ROTATION SCHEMES AS CYCLIC PERMUTATIONS

Ioannis N. Athanasiadis[1], Andrea E. Rizzoli[1], Sander Janssen[2] and Martin van Ittersum[2]
[1] Istituto Dalle Molle di Studi sull'Intelligenza Artificiale, Lugano, Switzerland, {andrea,ioannis}@idsia.ch
[2] University of Wageningen, The Netherlands, {sander.janssen,martin.vanittersum}@wur.nl

Introduction

Farm production planning involves the simulation and evaluation of alternative crop succession schemes, known also as crop rotation cycles. A crop rotation cycle is a sequence of crops that are applied cyclically on the same piece of land. Typically, artificial crop rotation schemes are generated as all possible rearrangements of the available crops that are subsequently filtered with respect to cyclic equivalence and crop succession suitability requirements. Given a set C of n crops and a desired length of rotations r, the traditional approach requires the evaluation of a solution space, sized n^r. This practice limits the length of rotations to be evaluated as the memory required for storing all crop rearrangements expands exponentially. In this paper, we present an alternative generation algorithm that excludes from the solution space all cyclically equivalent rotations. The algorithm represents each crop rotation cycle as a number in the n-based numeral system, and is capable of excluding the generation of cyclic equivalent rotations, through a single modulo operation. Two alternatives of the algorithm are presented: The first excludes the cyclic equivalent rotations of the same length (i.e. the maize – fallow – maize rotation is equivalent with the rotation maize – maize – fallow). The second variation of the algorithm also excludes the cyclic equivalent rotations of lesser orders (i.e. the maize – fallow – maize – fallow rotation of length four is equivalent with the rotation maize – fallow of length two).

Methodology

Let $C=\{c_0, c_1, \ldots, c_{n-1}\}$ be a set of n crops. A rotation of length r is an ordered sequence (rearrangement) of r elements of C. A rotation can be considered as a permutation with repetition of size r of the elements of C. The number of all possible sequences of crops is the number of permutations with repetition and equals to n^r. In general, each permutation can be identified by a unique sequence of r digits, which represents an integer number in the n-base numeral system. The actual value of a sequence of r digits "$d_r \cdots d_k \cdots d_2 d_1$" in the n-base system is given by the equation: $i=\sum_{k=1}^{r} d_k \cdot n^{k-1}$. As an example, the four-digit sequence 1012 in the 3-base system represents the decimal number $1\cdot3^3+0\cdot3^2+1\cdot3^1+2\cdot3^0=27+0+3+2=32$. Note that $d_k=i \text{ modulo } n^{k-1}$. In the case of having a set of $n=3$ crops $C=\{c_0,c_1,c_2\}$, and a rotation length $r=4$, the all possible permutations with repetition, and their representation in the 3-base numeral system are those illustrated in Table 1.

Table 1: Example rotations and their representations in the c-based and decimal numeral system

Rotation	Representation (index) in the 3-base system	Representation (value) in the decimal system
$[c_0,c_0,c_0,c_0]$	0000	0
$[c_0,c_0,c_0,c_1]$	0001	1
$[c_0,c_0,c_0,c_2]$	0002	2
$[c_0,c_0,c_1,c_0]$	0010	3
$[c_0,c_0,c_1,c_1]$	0011	4
$[c_0,c_0,c_1,c_2]$	0012	5
$[c_0,c_0,c_2,c_0]$	0020	6
…	…	…
$[c_1,c_0,c_1,c_2]$	1012	32
…	…	…
$[c_2,c_2,c_2,c_2]$	2222	$80 = 3^4-1$

Let $C=\{c_0,c_1,..c_{n-1}\}$ be a set of n crops, and r be the length of the permutations with repetition to be generated. Each permutation $[d_r \cdots d_3 d_2 d_1]$ of size r can be uniquely identified by a single integer $i=\sum_{k=1}^{r} d_k \cdot n^{k-1}$, where $i\geq0$, *and* $i<n^r$. Simply, by counting from 0 to $n^r\text{-}1$ in the n-base numeral system, all possible permutations with repetition of length r can be generated. In this way, a simple permutation generator with repetition can be specified, as a simple counter from 0 to $n^r\text{-}1$. However, for generating artificial crop rotation schemes, we need to produce all ***cyclic*** permutations with repetition. This means that in the abovementioned example, the rotation $\{12\}:[c_0,c_1,c_1,c_0]$ is equivalent with the rotation $\{4\}: [c_0,c_0,c_1,c_1]$, thus one of the two needs to be excluded from generation. We underline that every sequence of r digits $[d_r \cdots d_3 d_2 d_1]$ has at most $r\text{-}1$ cyclic equivalents, which can be produced by applying a shift function $r\text{-}1$ times, and they are: $\{[d_{r-1} \cdots d_3 d_2 d_1 d_r], [d_{r-2} \cdots d_3 d_2 d_1 d_r d_{r-1}], \ldots [d_1 d_r \cdots d_3 d_2]\}$. Note that the index of the sequence $[d_r \cdots d_3 d_2 d_1]$ is $i=\sum_{k=1}^{r} d_{k-m} \cdot n^{k-1}$. The index of the m-th cyclic equivalent is given by the form: $i_m=\sum_{k=1}^{r} d_{k-m} \cdot n^{k-1}$, where $m=1 \ldots r$.

Based on this remark, a generator of ***cyclic permutations with repetition*** can be defined as follows: Let C be a set of crops of size n, and r be the rotation length, then each permutation with repetition can be uniquely identified by a single integer i, where $i \geq 0$, and $i < n^r$. This rotation has (at most) r cyclic equivalents, with index i_m, where $m=1...r$. Based on the above, the generation of cyclic permutations with repetition can be achieved as follows:

Algorithm I: Exclude the cyclic equivalents of the same order

For $i \in [0, n^r)$, that represents a candidate rotation, check if there is at least one i_m, for $m=[1,r]$, such as $i_m < i$. If this condition is true, then there is at least one cyclic equivalent rotation of the same length generated for for an i' value smaller than the current i. Therefore the current rotation with id i should be be skipped. Else consider i as a rotation for which no cyclic equivalent has been produced before.

Algorithm II: Exclude the cyclic equivalents of the same or lesser orders

For each $i \in [0, n^r)$ evaluate if there is at least one i_m for $m=[1,r]$, such as $i_m(m) \leq i$. (The difference with the previous case is the equality condition). If this condition is true then there is at least one cyclic equivalent permutation of the same or lesser length has been already generated, thus do not consider the candidate rotation i. Else, generate i as a rotation for which no other cyclic rotation generated of the same of lesser order.

Results

This paper demonstrated how cyclic equivalent crop rotations could be excluded at generation phase, which affects significantly the volume of the problem space upon which crop succession suitability requirements filters need to be applied. The following table presents the size of all rotations to be evaluated for number of crops $c=2...9$ and for rotation length $r=2...9$, for all three cases: the conventional methods, that generates all permutations, and the two variations of the algorithm above (case 1 and case 2). By excluding from the solution space at generation time the cyclic equivalent rotations, the solution space is reduced up to by 90% (i.e. in the case of 8 crops and rotation length equal to 9 instead of more than 134 million rotations, it is sufficient to evaluate less than 15 million alternatives, as the rest 120 millions are cyclic equivalents). It becomes apparent that the proposed method reduces significantly the volume of the alternatives to be evaluated.

The application of agronomic suitability filters remains the same with the current approach as with the conventional ones. However, the volume of solution space is drastically reduced.

Table 2: Comparison of the solution space volume for the conventional method and the new two algorithms. Presented for various values of crop set size (c) and desired rotation length (r)

c	*r*	*CONV (n^r)*	*CASE 1*	*CASE2*	*c*	*r*	*CONV (n^r)*	*CASE 1*	*CASE2*
2	2	4	3	1	6	2	36	21	15
2	3	8	4	2	6	3	216	76	70
2	4	16	6	3	6	4	1,296	336	315
2	5	32	8	6	6	5	7,776	1,560	1,554
2	6	64	14	9	6	6	46,656	7,826	7,735
2	7	128	20	18	6	7	279,936	39,996	39,990
2	8	256	36	30	6	8	1,679,616	210,126	209,790
2	9	512	60	56	6	9	10,077,696	1,119,796	1,119,720
3	2	9	6	3	7	2	49	28	21
3	3	27	11	8	7	3	343	119	112
3	4	81	24	18	7	4	2,401	616	588
3	5	243	51	48	7	5	16,807	3,367	3,360
3	6	729	130	116	7	6	117,649	19,684	19,544
3	7	2,187	315	312	7	7	823,543	117,655	117,648
3	8	6,561	834	810	7	8	5,764,801	720,916	720,300
3	9	19,683	2,195	2,184	7	9	40,353,607	4,483,815	4,483,696
4	2	16	10	6	8	2	64	36	28
4	3	64	24	20	8	3	512	176	168
4	4	256	70	60	8	4	4,096	1,044	1,008
4	5	1,024	208	204	8	5	32,768	6,560	6,552
4	6	4,096	700	670	8	6	262,144	43,800	43,596
4	7	16,384	2,344	2,340	8	7	2,097,152	299,600	299,592
4	8	65,536	8,230	8,160	8	8	16,777,216	2,097,684	2,096,640
4	9	262,144	29,144	29,120	8	9	134,217,728	14,913,200	14,913,024
5	2	25	15	10	9	2	81	45	36
5	3	125	45	40	9	3	729	249	240
5	4	625	165	150	9	4	6,561	1,665	1,620
5	5	3,125	629	624	9	5	59,049	11,817	11,808
5	6	15,625	2,635	2,580	9	6	531,441	88,725	88,440
5	7	78,125	11,165	11,160	9	7	4,782,969	683,289	683,280
5	8	390,625	48,915	48,750	9	8	43,046,721	5,381,685	5,380,020
5	9	1,953,125	217,045	217,000	9	9	387,420,489	43,046,889	43,046,640

Acknowledgement

The work presented in this publication is funded by the SEAMLESS integrated project, EU 6th Framework Programme.

Bibliography

D.Knuth, Permutations and Factorials, 1997. The art of computer programming, 3rd edition, 45-50.

R. Sedgewick, Permutation Generation Methods, 1977. ACM Computing Surveys. 9, 137-164.

S. Dogliotti, W. Rossing, M. van Ittersum. ROTAT, a tool for systematically generating crop rotations, 2003. European Journal of Agronomy. 19, 239-250.

W.K.Haneveld, A. Stegeman. Crop succession requirements in agricultural production planning, 2005. European Journal of Operational Research. 166, 406-429.

N. K. Detlefsen, A.L. Jensen. Modelling optimal crop sequences using network flows, 2007. Agricultural Systems. 94, 566–572.