Université Libre de Bruxelles

Institut de Recherches Interdisciplinaires et de Développements en Intelligence Artificielle

IRIDIA

Search space for AutoMoDe-`Chocolate` and AutoMoDe-`Maple`

J. Kuckling, A. Ligot, D. Bozhinoski, and M. Birattari

IRIDIA – Technical Report Series

Technical Report No.
TR/IRIDIA/2018-012

December 2018

IRIDIA – Technical Report Series
ISSN 1781-3794

Published by:
IRIDIA, *Institut de Recherches Interdisciplinaires et de Développements en Intelligence Artificielle*
UNIVERSITÉ LIBRE DE BRUXELLES
Av F. D. Roosevelt 50, CP 194/6
1050 Bruxelles, Belgium

Technical report number TR/IRIDIA/2018-012

Search space for AutoMoDe-`Chocolate` and AutoMoDe-`Maple`

Jonas Kuckling, Antoine Ligot
Darko Bozhinoski, and Mauro Birattari

IRIDIA, CoDE, Université Libre de Bruxelles,
Brussels, Belgium.

December 14, 2018

1 Introduction

AutoMoDe, an abbreviation for automatic modular design, is a class of design methods that combines predefined modules into control software for robot swarms (Francesca et al., 2014). In this document, we analyze the size of the search space of two AutoMoDe instances: `Chocolate` (Francesca et al., 2015), and `Maple` Kuckling et al. (2018). `Chocolate` generates finite state machines that comprise up to four states, and up to four outgoing transitions per states. `Maple` generates behavior trees. The architecture of the generated behavior trees is restricted to a fixed top-level sequence* ($\rightarrow^*$) node that has up to four subtrees as children. Each subtree contains a selector (?) node with exactly two children: one condition node and one action node.

We do not take any symmetry or similarity into account. We will represent the search space of all possible behaviors and conditions through two sets $\mathcal{B}$ and $\mathcal{C}$. These sets describe all possible combinations of a behavior or condition with all possible parameters.

2 Finite-state machines

General case

Consider a finite-state machine with up to s_{max} states. Each state has at least one and at most t_{max} outgoing transitions that cannot point back into the same state they origin from. The number of all such finite-state machines (written $|S_{FSM}|$) can be described by equation 1:

$$\begin{aligned} |S_{FSM}| &= \sum_{s=1}^{s_{max}} |S_{FSM}(s)| \\ &= |S_{FSM}(1)| + \sum_{s=2}^{s_{max}} |S_{FSM}(s)|. \end{aligned} \tag{1}$$

$|S_{FSM}(s)|$ describes the number of possible finite-state machines with exactly s states. The trivial case of $s = 1$ needs to be handled separately as a finite-state machine with exactly one state does not have any transitions. There are exactly $|S_{FSM}(1)| = |\mathcal{B}|$ finite-state machines with exactly one state, as only the behavior for the single state can be chosen. For more than one state $s > 1$, the number of possible finite-state machines can be described by the number of s (independent) choices, one for each state. This can be modeled by choosing s times independently from the set of all possible states (*multiplication principle*). In this model, the state already contains information about all outgoing transitions (e.g. their number, the target, and the condition).

$$|S_{FSM}(s)| = |S_{state_1}(s)| * |S_{state_2}(s)| * ... * |S_{state_s}(s)| = \prod_{i=1}^{s} |S_{state}(s)|. \tag{2}$$

If the number of states in a finite-state machine is fixed to s, then number of configurations for each state is the same, and can be expressed by $|S_{state}(s)|$. This is because each state is composed of the behavior (chosen from $\mathcal{B}$) and up to t_{max} outgoing transitions to the $s - 1$ other states.

$$|S_{state}(s)| = \sum_{t=1}^{t_{max}} \left(|\mathcal{B}| * \prod_{j=1}^{t} |S_{transition}(s)| \right). \tag{3}$$

The number of possible outgoing transition is defined by the (independent) choice of a condition (from the set $\mathcal{C}$) and the target state. The target state can be modeled by a mapping from the states unto the set $\{1, 2, ..., s-1\}$. Therefore there are $s - 1$ possible target states for the transitions.

$$|S_{transition}(s)| = |\mathcal{C}| * (s - 1). \tag{4}$$

By substituting the equation for the number of outgoing transitions of a state (equation 4) into the equation for the number of possible configurations of a state (equation 3) we obtain:

$$\begin{aligned} |S_{state}(s)| &= \sum_{t=1}^{t_{max}} \left(|\mathcal{B}| * \prod_{j=1}^{t} (|\mathcal{C}| * (s-1)) \right) \\ &= \sum_{t=1}^{t_{max}} \left(|\mathcal{B}| * |\mathcal{C}|^t * (s-1)^t \right). \end{aligned} \tag{5}$$

Substituting the result obtained in equation 5 into equation 2 then leads to:

$$\begin{aligned} |S_{FSM}(s)| &= \prod_{i=1}^{s} \sum_{t=1}^{t_{max}} \left(|\mathcal{B}| * |\mathcal{C}|^t * (s-1)^t \right) \\ &= \left(\sum_{t=1}^{t_{max}} \left(|\mathcal{B}| * |\mathcal{C}|^t * (s-1)^t \right) \right)^s . \end{aligned} \tag{6}$$

Finally after substituting equation 6 into equation 1, we obtain an equation for the size of the search space for all finite-state machines with up to s_{max} states and t_{max} outgoing transitions per state:

$$|S_{FSM}| = |\mathcal{B}| + \sum_{s=2}^{s_{max}} \left(\sum_{t=1}^{t_{max}} \left(|\mathcal{B}| * |\mathcal{C}|^t * (s-1)^t \right) \right)^s . \tag{7}$$

Chocolate

In `Chocolate` the finite-state machines are limited to four states and each state can only have up to four outgoing transitions. Inserting the values $s_{max} = 4$ and $t_{max} = 4$ into equation 7 leads to the following equation:

$$
\begin{aligned}
\#S(FSM) = {} & 43046721\,|\mathcal{B}|^4\,|\mathcal{C}|^{16} + 57395628\,|\mathcal{B}|^4\,|\mathcal{C}|^{15} + 47829690\,|\mathcal{B}|^4\,|\mathcal{C}|^{14} \\
& + 31886460\,|\mathcal{B}|^4\,|\mathcal{C}|^{13} + 16474671\,|\mathcal{B}|^4\,|\mathcal{C}|^{12} + 7085880\,|\mathcal{B}|^4\,|\mathcal{C}|^{11} \\
& + 2598156\,|\mathcal{B}|^4\,|\mathcal{C}|^{10} + 787320\,|\mathcal{B}|^4\,|\mathcal{C}|^{9} + 203391\,|\mathcal{B}|^4\,|\mathcal{C}|^{8} \\
& + 43740\,|\mathcal{B}|^4\,|\mathcal{C}|^{7} + 7290\,|\mathcal{B}|^4\,|\mathcal{C}|^{6} + 972\,|\mathcal{B}|^4\,|\mathcal{C}|^{5} + 81\,|\mathcal{B}|^4\,|\mathcal{C}|^{4} \\
& + 4096\,|\mathcal{B}|^3\,|\mathcal{C}|^{12} + 6144\,|\mathcal{B}|^3\,|\mathcal{C}|^{11} + 6144\,|\mathcal{B}|^3\,|\mathcal{C}|^{10} + 5120\,|\mathcal{B}|^3\,|\mathcal{C}|^{9} \\
& + 3072\,|\mathcal{B}|^3\,|\mathcal{C}|^{8} + 1536\,|\mathcal{B}|^3\,|\mathcal{C}|^{7} \\
& + 640\,|\mathcal{B}|^3\,|\mathcal{C}|^{6} + 192\,|\mathcal{B}|^3\,|\mathcal{C}|^{5} + 48\,|\mathcal{B}|^3\,|\mathcal{C}|^{4} + 8\,|\mathcal{B}|^3\,|\mathcal{C}|^{3} + |\mathcal{B}|^2\,|\mathcal{C}|^{8} \\
& + 2\,|\mathcal{B}|^2\,|\mathcal{C}|^{7} + 3\,|\mathcal{B}|^2\,|\mathcal{C}|^{6} + 4\,|\mathcal{B}|^2\,|\mathcal{C}|^{5} + 3\,|\mathcal{B}|^2\,|\mathcal{C}|^{4} + 2\,|\mathcal{B}|^2\,|\mathcal{C}|^{3} \\
& + |\mathcal{B}|^2\,|\mathcal{C}|^{2} + |\mathcal{B}|
\end{aligned}
\tag{8}
$$

In this equation, each summand $x\,|\mathcal{B}|^b\,|\mathcal{C}|^c$ indicates the number of possible finite-state machines with b states and a total of c transitions. The term x denotes the number of possible finite state machines that combine c transitions and b states with the restriction of at most t_{max} outgoing transitions per state.

For example $2\,|\mathcal{B}|^2\,|\mathcal{C}|^7$ denotes that `Chocolate` can generate 2 different topologies of finite-state machines containing 2 states and 7 outgoing transitions. Because of the limit of $t_{max} = 4$ outgoing transitions per state, 3 of the 7 outgoing transitions need to be associated to one state, and the 4 remaining transitions to the other state. However each of the two states can have the four transitions, leading to two possible distributions.

Summing the coefficients of equation 8 results in a total of 207 387 017 different topologies of finite-state machines.

The defining factor for the size of the search space is however the search space defined by the modules. Even without taking the parameters into account, there are $6^{16} = 2\,821\,109\,907\,456$ possible ways of assigning a condition to each transition in the case of the maximum number of states and transitions. We can therefore conclude that the size of the search space for `Chocolate` is in $O\left(|\mathcal{B}|^4\,|\mathcal{C}|^{16}\right)$.

3 Behavior tree

General case

Consider a behavior tree with depth d, that is, $d+1$ nodes on the longest path from the (implicitly) defined root node to a leaf node. Additionally, let the top-level node (only child of the root), be at level 1. All of its children are at level 2, their children at level 3, and so on. In this case the level of a node is equivalent to its depth in the tree.

Suppose that we fix a level i. On this level i we can choose either a control-flow node out of a subset of all possible control-flow nodes $\mathcal{N}_i \in \mathcal{N}$, or a leaf node (either action or condition node). Additionally, every control-flow node on level i must have between c_{min} and c_{max} children.

Let $BT_{=l}$ be the set of behavior trees with a depth of exactly l. That is the there are exactly l nodes from the top-level node to the furthest leaf node.

Similarly let $BT_{<l}$ be the set of behavior trees where there exists no path between the top-level node and any leaf node that has at least l nodes in it. It should be noted that the following equality holds true:

$$BT_{<l} = \bigcup_{i=1}^{l-1} BT_{=i}. \tag{9}$$

The last important notation is $BT_{\leq l}$, the set of all behavior trees with a depth of at most l. The following two equalities hold up:

$$BT_{\leq l} = BT_{<l+1} \tag{10}$$

$$BT_{\leq l} = BT_{=l} \cup BT_{<l}. \tag{11}$$

The number of behavior trees with at most l levels can be described by the following recursive formula:

$$|BT_{=1}| = |\mathcal{B}| + |\mathcal{C}| \tag{12}$$

$$\begin{aligned} |BT_{\leq l+1}| &= |BT_{=l+1}| + |BT_{<l+1}| \\ &= |BT_{=l+1}| + |BT_{\leq l}| \\ &= \sum_{i=l+1}^{1} |BT_{=i}| \,. \end{aligned} \tag{13}$$

It should be noted that this formula covers the recursive anchor for $l = 1$ (the leaf nodes). If the restrictions applied to a behavior tree allow it, this recursive formula can also have a recursive anchor for $BT_{=i}, i > 1$

For $|BT_{=i}|\,, i > 1$ we can show the following:

$$|BT_{=i}| = |\mathcal{N}_i| \sum_{c=c_{min}}^{c_{max}} \left(|BT_{=i-1}| + |BT_{<i-1}|\right)^c - |BT_{<i-1}|^c \,. \tag{14}$$

That is because no behavior trees with $i > 1$ levels can be a leaf node. Additionally they if they have a depth of i, they need to have at least one subtree under the top-level node with exactly $i-1$ levels. It is however not necessary to only have a single subtree with these many levels. Indeed any number of subtrees (with at least more than one) are acceptable. By the inclusion-exclusion principle, we can include all behavior trees as subtrees that have either $l-1$ or less then $l-1$ levels ($(|BT_{=i-1}| + |BT_{<i-1}|)^c$) but we need to include the case that all subtrees have less then $l-1$ levels ($|BT_{<i-1}|^c$). This needs to be done for all mutually exclusive choices for the number of children and all independent choices of the control-flow node.

Maple

In `Maple` we have a restricted version of the behavior trees, that can have exactly three levels. Because of the special restrictions, we can define a recursive anchor for $BT_{=2}$, describing our selector subtrees. There are $|\mathcal{C}| * |\mathcal{B}|$ possible combinations for the selector subtrees, because of the independent choices of the selector node (no true single choice), condition for the condition node and behavior for the action node.

$$\begin{aligned} |BT_{=2}| &= |\{?\}| * |\mathcal{C}| * |\mathcal{B}| \\ &= |\mathcal{C}| * |\mathcal{B}| \,. \end{aligned} \tag{15}$$

Additionally all other levels (in this case only the top-level) can have between $c_{min} = 1$ and $c_{max} = 4$ children. If we use these restrictions in equation 14 it results in:

$$\begin{aligned} |BT_{=3}| = |\mathcal{N}_3| \sum_{c=c_{min}}^{c_{max}} (|BT_{=2}| + |BT_{<2}|)^c - |BT_{<2}|^c \\ &= |\{\rightarrow^*\}| \sum_{c=1}^{4} (|BT_{=2}| + 0)^c - 0^c \\ &= \sum_{c=1}^{4} (|BT_{=2}|)^c \\ &= \sum_{c=1}^{4} (|\mathcal{C}| * |\mathcal{B}|)^c \\ &= |\mathcal{C}|^1 |\mathcal{B}|^1 + |\mathcal{C}|^2 |\mathcal{B}|^2 + |\mathcal{C}|^3 |\mathcal{B}|^3 + |\mathcal{C}|^4 |\mathcal{B}|^4 \,. \end{aligned} \tag{16}$$

Here again the coefficients of $x\, |\mathcal{C}|^i\, |\mathcal{B}|^i$ describe the number of ways it is possible to construct a restricted behavior tree with i. However there is just a single way of combining the subtrees (all under the top-level node).

4 Conclusion

As shown above the size of the search space of the finite state machines is in $O(|\mathcal{B}|^4\, |\mathcal{C}|^{16})$ while the size of the search space for the behavior trees is in $O(|\mathcal{B}|^4\, |\mathcal{C}|^4)$.

The large factors in the calculation for the finite state machines (see equation 8) are also small when compared to the combination of possible modules, because $|\mathcal{C}| \geq 600$ (6 conditions with at least one parameter p that can take up to 100 values) and therefore $|\mathcal{C}|^{16} \geq 2.821109907456 \times 10^{44}$ and even $|\mathcal{C}|^4$, one of the defining factors for the search space of the behavior trees, is already larger than the constants: $|\mathcal{C}|^4 \geq 1.296 \times 10^{11}$. Both of these numbers are considerably bigger than the factors in the equation 8.

Arguably this comparison is not entirely fair as changes in parameters probably have smaller effects on the performance than the placement of a single transition.

Acknowledgements.

The project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 681872). Mauro Birattari and Jonas Kuckling acknowledge support from the Belgian *Fonds de la Recherche Scientifique* – FNRS.

References

Francesca, G., Brambilla, M., Brutschy, A., Garattoni, L., Miletitch, R., Podevijn, G., Reina, A., Soleymani, T., Salvaro, M., Pinciroli, C., and Birattari, M. (2015). AutoMoDe-Chocolate: automatic design of control software for robot swarms. *Swarm Intelligence*, 9(2/3):125–152.

Francesca, G., Brambilla, M., Brutschy, A., Trianni, V., and Birattari, M. (2014). AutoMoDe: A novel approach to the automatic design of control software for robot swarms. *Swarm Intelligence*, 8(2):89–112.

Kuckling, J., Ligot, A., Bozhinoski, D., and Birattari, M. (2018). Behavior trees as a control architecture in the automatic modular design of robot swarms. In Dorigo, M. and et al., editors, *Swarm Intelligence, ANTS*, volume 11172 of *LNCS*, pages 30–43. Springer, Cham, Switzerland.